Large Eddy Simulation in internal combustion Engine

Vom Fachbereich Maschinenbau
an der Technischen Universität Darmstadt
zur
Erlangung des Grades eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

vorgelegt von

Dipl.-Ing. Matteo Ghelfi

aus Ferrara (Italien)

Berichterstatter:	Prof. Dr.-Ing. J. Janicka
Mitberichterstatter:	Prof. Dr. J. Lang
Tag der Einreichung:	30. April 2013
Tag der mündlichen Prüfung:	12. Juni 2013

Darmstadt 2013
D17

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbstständig verfasst und keine anderen als die von mir angegebenen Hilfsmittel verwendet habe. Ich erkläre außerdem, dass ich bisher noch keinen Promotionsversuch unternommen habe.

Matteo Ghelfi
Darmstadt, den 30.04.2013

Contents

List of Symbols

Latin Symbols, Upper Case

B_{ij},A_{ij}	Preexponential Factors
C_s	Smagorinsky model constant
S_{ij}	Resolved shear stress tensor components
T_{aj}	Activation Temperature
V	Volume
C	Germano model factor
D	Diffusion Term
E	Activation Energy
F	Thickening Factor
I	Internal Energy
J	Heat Flux Vector
M	Pope verification parameter
Q	Source Term in energy equation
R	Universal Gas Constant
S	Source Term
T	Temperature
u’	Subgrid scale velocity
W	Molecular Weight
X	Mole Fraction
Y	Mass Fraction

Latin Symbols, Lower Case

a_{ij}, b_{ij}	Stoichiometric coefficients
s_L	Laminar Flame Velocity
v_{avg}	Mean velocity
v_{rms}	Mean velocity fluctuation
g	Gravitational Force
m	Mass
t	Time
u	Velocity on x component
v	Velocity on y component
w	Velocity on z component

Greek Symbols, Upper Case

Δ	Filter Dimension
Γ	Wrinkling factor calculation function

λ	Termal Conductivity
Ω	Flame sensor
ω	Reaction Rate
Φ	General Scalar
σ	Stress tensor
Θ	Reduced Temperature
Ξ	Wrinkling factor
h	Hentalpy

Greek Symbols, Lower Case

α	Thermal diffusivity rate
β	Temperature exponent
δ_L^0	Flame Thickness
δ_{ij}	Kronecker Delta
$\dot{\rho}$	Source Term in scalar transport equation
ϵ	Turbulence Disipation rate
η	Lenghtscale
ν	Kinematic viscosity
ν_t	Subgrid scale viscosity
Ω	Progress Variable
ϕ	Mixture fraction
ρ	Density
τ	Stress tensor
τ_η	Timescale
ϱ	Density

Operators and Symbols

$\overline{\Phi}$	Reynolds Filter
$\widehat{\Phi}$	Germano test factor
$\widetilde{\Phi}$	Favre Filter

Abbreviations

BDC	Bottom Death Center
CA	Crank Angle
CFD	Computational Fluid Dynamic
IC	Internal Combustion
LES	Large Eddy Simulation
RANS	Reynolds Averaged Navier-Stokes
SI	Spark Ignited
TDC	Top Death Center
0-D	Zero Dimensional
1-D	One Dimensional
2-D	Two Dimensional
3-D	Three Dimensional
ATDCE	After Top Death Center Exhaust Stroke

ATF	Artificially Thickened Flame
CFL	Courant-Friedrichs-Lewy
PIV	Particle Imaging Velocimetry
rms	Root Mean Square
sgs	Subgrid scale
M	Million

Dimensionless Numbers

Re_t	Reynolds Turbulent Number
Da	Damkoeler Number
Pr	Prandtl Number
Re	Reynolds Number
Sc	Schmidt Number

Chapter 1

Introduction

Fossil fuel combustion is the most used source of energy production in modern power generation. The development of other kind of energy sources is always an important theme for present and future, but alternative energy sources produce less than 20% of the total world energy production.

Possible alternative sources could be solar and wind, which had in the last years a strong increment but those resources are still more expensive than fuel combustion. Nuclear energy is an other kind of power production, but after Fukushima 2011 strong action against this kind of resource reduced drastically the amount of production in some countries like Germany, where all nuclear power plan will be dismissed in the next few years.

Fossil fuel energy is also the center of discussion because of pollutant formation. In particular, the most discussed and subject of strong reduction is the carbon dioxide (CO_2) that leads to green-house effect. Other controlled and reduced pollutants are nitrogen oxides NO_x, carbon monoxide and particulates.

Automotive sector is one of the most strongly afflicted from pollutant reduction. This kind of industry is concentrating a large part of resources in research to develop new methods of emission reduction and control. Therefore, it is really important to develop new analysis system to understand better physical phenomena that lead to pollutants formation.

The objective of this work is to initiate the development for a future method that can analyze and reduce pollutant creation in the combustion phase inside engine. This can be used during engine design and development, reducing prototyping and calibration costs.

In particular, this work will concentrate in computational fluid dynamic (CFD) analysis of turbulent combustion processes in internal combustion (IC) engine using large eddy simulation (LES), which is on first stage of development and still not used by engine companies.

Objective of this Work

Within few years, when computational power will increase at least five or ten times, LES method will become a standard procedure for engineering development. Of course, IC engines will be studied in this context. Therefore, in the present time the process of development of robust and precise simulation method is already started.

This work focuses on the development of the method which in the future can become routine for every IC engine company. However, the method is at initial stage of the

development. The scientific community studies LES to understand if it can be used, taking into account outcome quality and application facility.
Therefore this work focuses on easy handling models implementation and explains how to use a graphic mesh generation software to create a feasible computational domain.
Besides, the chosen software KIVA-4MPI needs to be improved which scope is to use it easily during the widest range of simulation.

State of the Art

In the present time IC engine companies understand that CFD is a really important phase of the engine development, therefore it is often adopted during the process.
CFD is generally used to study and develop the intake and exhaust channel. In particular giving them the shape to bring inside the cylinder the perfect flow structure and turbulence needed to help combustion process reaching the optimal efficiency. Besides, CFD is also used to study the shape of the valves and their movement. An other important simulation done by CFD is the heat release verification, to verify if the cooling system is working well and the engines will never overheats. In-cylinder CFD simulation are also used for injection and air charging phases, when it is important to take care about the larger flow structure and prepare the air/fuel mixture to combustion phase.
At present time the majority CFD simulation are done by Reynolds Average Navier-Stokes (RANS) method. This is available for every CFD commercial software and does not require a large amount of cells. The small structures creating turbulence are modeled. Due to the complexity of the engine system it is easy to reach more than a million control volumes. This kind of computational domain calculated by RANS simulation requires a lot of computational power to be solved. If the same simulation will be done using LES method the number of cells will be higher and the computational time increases cubically.
Large eddy simulation method is a powerful instrument, which is developed since '90. In the present time it is the most studied method on the scientific community. Applications and new models are presented on every CFD conference. The actual computational available power allows to use this method in research projects.
With LES development a new kind of application can be studied, the combustion for 3-D cases. Combustion is a complex topic to be studied with RANS because of the averaged nature of the method, which disagree with the high instationary nature of combustion process. It is easier to include combustion in LES context.
Due to the instationary nature of LES coupled with combustion, computational time is increasing by the large amount of requested control volumes and by the high physical time simulated. This second point is necessary to obtain statistical results.
In-cylinder IC engine simulation is a particular case of CFD simulation. The domain is moving and a complete simulation needs to include all the phases during the engine cycle. Therefore, standard LES of an IC engine simulates more physical time than a steady geometry simulation. Statistic collection in a IC engine is done collecting data on one certain crank angle (CA) position for each cycle. So a data set needs at least fifty computed cycles to be consider a robust outcome.

Overview of this Work

This work will focus on the theory and the method used to develop the software and prepare it for IC engine simulation.
The first part of the work is to understand the processes included inside internal combustion engine simulation coupled with the combustion physics. Therefore, the Chapter 2 gives a IC engine introduction including the technologies which have stronger influence on the combustion simulation. Afterward in the Chapter 3 the physical phenomena of basic flow and combustion are explained, with a particular focus on the main topic IC engines.
A time consuming part of the work is the introduction and development of the models necessary to have a quality simulation of IC engine in LES context. At the beginning of the work the used software was not ready to give quality results of a multicycle simulation in LES context including combustion. Besides, a complete validation of the software supported by experimental result was not done.
The Chapter 4 and 5 focus on modeling and numerical procedure necessary to increase the simulation quality. The objective of the improvement is to have a IC engine simulation with good match compared with experiments.
The implemented models need to be tested on a simulation and compared with other software. The elaborated software is verified and validated running a simple case before start with more complex simulation. Therefore, Chapter 6 focuses on the validation of the software and its models through a simple test case good documented by experiments and other software simulations.
When the software validation is done IC engine can be simulated. However, before simulation is presented a last time consuming topic is introduced in Chapter 7, mesh generation and management. This is a very important topic for this work, because in every moment of the simulation the grid quality is fundamental in LES context. Therefore, during the work in collaboration with T. Breitenberger and K. Nishad of Energie -und Kraftwerkstechnik institute on Darmstadt University of Technology a new method for grid generation compatible with KIVA-4MPI and LES was developed. Besides, the author developed and implemented a new grid motion method to keep quality grid during all the phases of the simulation.
The Chapters 8 and 9 focus on IC engine simulations. Two simulations are presented the first one of a two parallel valves engine and the second one of a four canted valves engine experimental studied by E. Baum and B. Peterson of Reaktive Strömungen + Messtechnik institute on Darmstadt University of Technology.
As last Chapter a briefly outlook of the work is given, resuming the results and opening more development possibilities for future works.

Chapter 2

Internal combustion engines

This first chapter has the objective to introduce the main topic of this work: internal combustion engines. The complexity of this topic are shown in several books. So,this chapter focuses on the principal aspects, which for the basis of the present study, phenomena that alter in-cylinder combustion. In particular, the newest technologies are treated, which are adopted by the most important engines companies.

2.1 Characterization

Internal combustion engines can be classified in gasoline or diesel engines, which are based on two different thermodynamic cycles. Both types of engines can be either two or four strokes depending on the air charge system.
Diesel engines are much heavier than gasoline engines in relation to developed power, which is caused by higher pressure and temperature to reach auto-ignition point. Therefore those engine parts must be developed for other kind of conditions distinguishing in bigger and more massive. Because diesel combustion is slower than gasoline combustion, this kind of engines reach slower regimes, which cause a reduction in power per cubic centimeter displacement.As a last disadvantage, diesel combustion creates more vibration than gasoline combustion and the engine leads to more noisy.
On the other hand diesel engines have better efficiency in wide range of regimes.

2.2 Important parameter

Most important parameter used to characterize IC engine are:

bore D is the cylinder diameter

stroke S double of crankshaft radius r_c or distance between bottom death center (BDC) and top death center (TDC) coordinate

displacement V is the total volume variation inside the cylinder during one stroke, written as:

$$V = \frac{\pi D^2 S}{4} \tag{2.1}$$

compression rate r rate between BDC and TDC volumes:

$$r = \frac{(V + V_c)}{V_c} \tag{2.2}$$

crankshaft velocity ω crankshaft rotation velocity, calculated with rps average velocity:

$$\omega = 2\pi n \tag{2.3}$$

crank angle CA instantaneous crankshaft position measured with angle from TDC:

$$\theta = 2\pi n(t - t_0) \tag{2.4}$$

where t is the instantaneous time and t_0 is the reference time

piston instant position s_p is the piston position in every moment. Supposing L_b the connecting rod length and $\Delta = r_c/L_b$:

$$s_p = \frac{C}{2}\left\{1 + \frac{1}{\Delta} - \cos\theta - \frac{1}{\Delta\sqrt{1 - \Delta^2\sin^2\theta}}\right\} \tag{2.5}$$

piston instant velocity u_p is the derivative of piston instantaneous position, of course on TDC and BDC is zero. Considering the average piston velocity as $\overline{u_p} = 2Cn$:

$$u_p = \overline{u_p}\frac{\pi}{2}\left\{\sin\theta + \frac{\Delta\sin 2\theta}{2\sqrt{1 - \Delta^2\sin^2\theta}}\right\} \tag{2.6}$$

piston instant acceleration a_p calculated from piston velocity with approximation as in [17], [12] and [28]:

$$a_p \approx \omega^2\frac{C}{2}[\sin\theta + \Delta\cos\theta] \tag{2.7}$$

instant combustion chamber volume V is the instantaneous available volume defined by [17] and [12]:

$$V_\theta = V\left\{\frac{1}{r-1} + \frac{1}{2}\left(1 + \frac{1}{\Delta} - \cos\theta - \frac{1}{\Delta}\sqrt{1 - \Delta^2\sin^2\theta}\right)\right\} \tag{2.8}$$

In Figure 2.1 the normalized rate of the functions, which is defined above are plotted. One can be seen which are the most critical point for a simulation, that are where piston velocity are larger, because a constant time step leads to larger piston displacement (cells inversion danger). Furthermore acceleration is higher due to pressure gradient which is large (instabilities due to high velocity), because volume function is similar to displacement function and volume variation will be similar to velocity function.

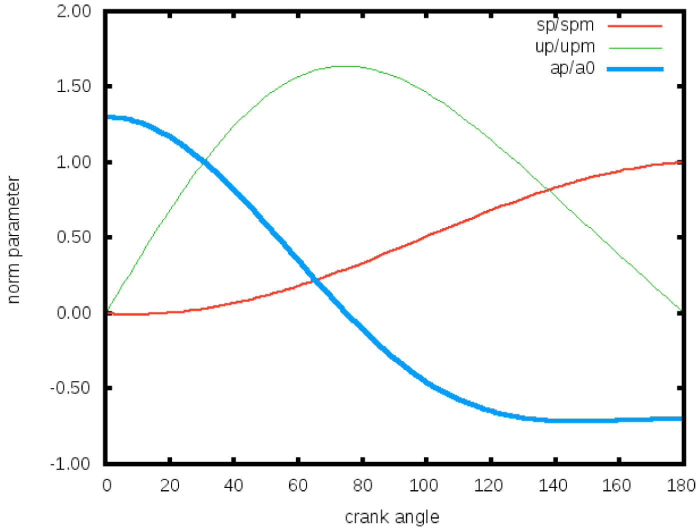


Figure 2.1: Instantaneous values of normalized parameters: piston displacement, velocity and acceleration calculated for $\Delta = 0.3$

2.2.1 Indicated work per cycle

Really important parameters since long time measurement to determine the engine performances, are the indicated parameter. First one is indicated work per cycle is obtained by pressure measurement, supposing the constant in the whole combustion chamber, inside the cylinder and plotted in $p-V$ diagram. Standard definition of indicated work is define by pressure integrated over the volume:

$$W_{c,i} = \oint pdV \tag{2.9}$$

While this parameter depends on the engine size is better to define an independent variable, which is the mean effective pressure, mean pressure that operate on the piston during the whole stroke. Integration over expansion strokes gives the results will be the indicated work of the entire cycle:

$$p_{me} = \frac{L_i}{V} = \frac{1}{V}\oint pdV \tag{2.10}$$

Furthermore can be calculated the efficiency of indicated cycle, which is defined as rate between indicated power and heat created from the reactions. First the indicated power is defined as [17],[7]:

$$P_i = \frac{dL_i}{dt} = L_i f_c = L_i \frac{n}{\epsilon} = p_{me} V \frac{n}{\epsilon} \tag{2.11}$$

whereas:
f_c is the cycle frequency
n is the value revolution per second (rps)
ϵ parameter that indicate 2 strokes ($\epsilon = 1$) or 4 strokes engine ($\epsilon = 2$)
onsidering $\dot{m}_c$ fuel total mass in cylinder and H_i heating value of fuel, the efficiency is

defined by:

$$\eta_i = \frac{P_i}{\dot{m}_c H_i} \tag{2.12}$$

which is also known as fuel conversion.

2.3 Combustion in spark ignited engines

Traditional spark ignited (SI) engine is charged with air and fuel premixed. The charge is quite homogeneous when the combustion starts. In normal operation the combustion starts with the spark plug that activates in the range of $40° \div 10° CA$ before top death center (BTDC).
After ignition the flame propagates gradually from the plug point through the combustion chamber volume, increasing pressure and temperature. Three different phase of combustion are:

1. flame development, where flame is created from the plug, the pressure does not increase significant at this time and chemistry is still slow;
2. turbulent combustion or flame propagation, the flame propagates through combustion chamber, pressure and temperature increase to maximum values. This phase may take $25° \div 30° CA$, whereas chamber volume shows small variation.
3. combustion completion, the flame front reaches walls and the flame extinguishes, pressure and temperature fall, volume increases quick and power is transferred to the piston.

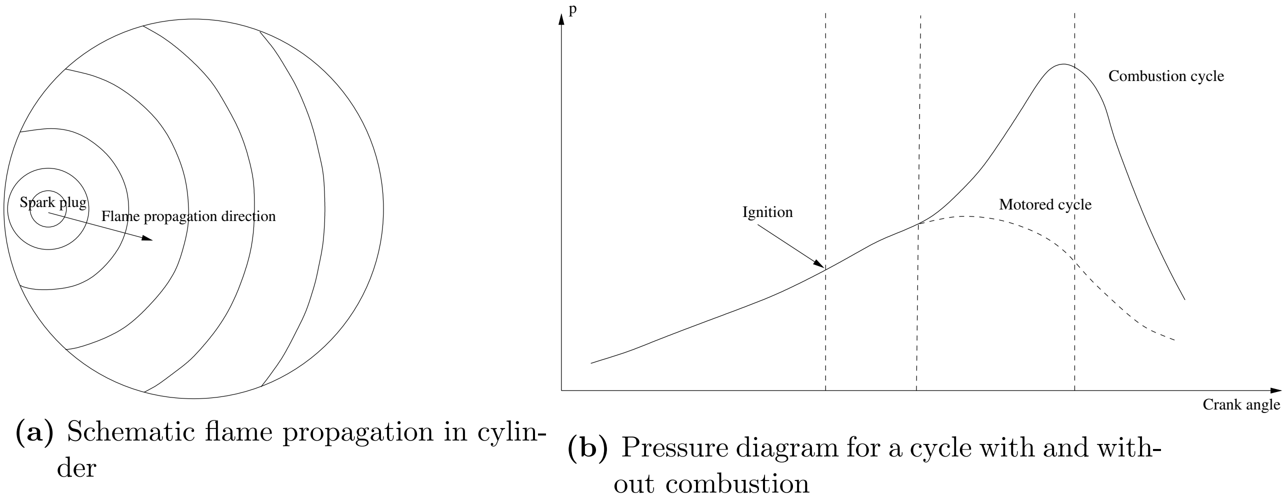
Those three phases are visible in Figure 2.2b, where pressure is plotted over crank angle. After ignition there is some lag before cylinder pressure starts to increase from the motored case distribution, when the separation happens phase one finishes and starts second phase until circa pressure peak, where the third part starts.

Further informations about combustion physics and mechanism will be presented in the next Chapter.

2.4 Direct injection engines

SI engines few years ago were the typical example of perfect premixed combustion, because standard system was port injection. In last years, after the introduction of auto ignition engines, direct injection was adopted to SI engines. This different concept allows very lean combustion (stratified charge) which:

1. reduces significantly pumping work at low load because of the lower pressure level in exhaust stroke and the possibility to neglect throttle plate;
2. reduces fuel consumption due to the best control of injected fuel in low regimes;



(a) Schematic flame propagation in cylinder

(b) Pressure diagram for a cycle with and without combustion

Figure 2.2: Combustion in cylinder phenomena

3. reduces exhaust emissions and allows better control.

GDI system's pressure is smaller than diesel injection, standard value is about $10 MPa$. The system construction results easier, lighter and cheaper than diesel's one. But physics of injection and combustion is different. For an engine construction company is not easy to change from port injection to GDI.
Phenomena that must be considered are:

1. spray-wall interaction, with piston and head configuration changes to optimize mixing;
2. spray-flow interaction to have a good mixing before ignition;
3. stratified charge formation and combustion;
4. injected fuel regulation in different regimes.

All those new concept are not easy to apply, because experiments for those studies are difficult to create. Simulation development and analysis started to be necessary to simulate what the experiment could not show.

2.5 Turbo charging and Downsizing

Turbo charging engine means to charge combustion chamber with compressed air, which increases the mass that can be burn in each cycle. Figure 2.3 shows compressor coupled with turbine disposed at exhaust port. The energy which is transfered to the flow comes directly from the burned gas. Regarding to equation 2.11:

$$P_e = V p_{me} \frac{n}{\epsilon} \tag{2.13}$$

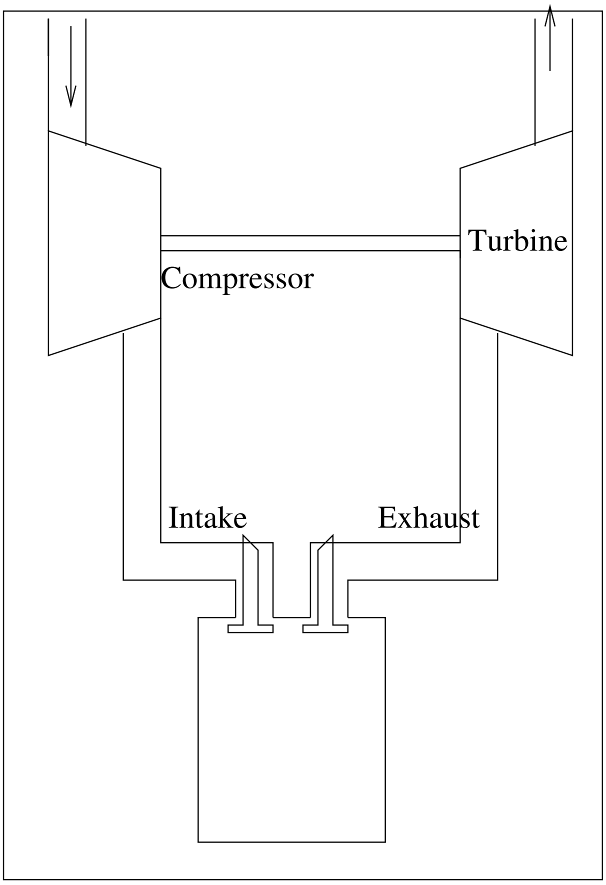


Figure 2.3: Simple scheme of turbo compression system

with same displacement engine power can be increased to run faster, but this will lead to problem with momentum of moving parts. An other option is increasing p_{me}. Of course, the construction must be robuster than without turbocharger but where force is increasing linear with pressure, inertia increases quadratically with piston mean velocity.
Diagram of Figure 2.4 shows a theoretical thermodynamic cycle, how compression can increase mean effective pressure. Cycle 1-2-3-4 is standard cycle without compressed aspiration, compressor increments point 1 pressure until point $1'$, during compression there is no significant difference between two cycles. At point $2'$ when combustion starts more heat is released than before, because of the increased fuel mass burned. Instead point 3, $3'$ is reached during combustion. After expansion physical properties are on point 4'. Therefor calculating p_{me} with indicated work shows a large increment.
Otherwise, it must be considered that the energy used for compression come from thermodynamic status of point 4'. Meanwhile standard cycle the residual energy of point 4 is wasted in exhaust. Turbo charged cycle part of this energy is used to increase intake pressure, so there is also an increment of thermodynamic efficiency.
Resuming, turbo charging has following advantages:

1. increases engine power or keeps constant by power reducing engine dimensions;
2. improves engine efficiency reducing specific fuel consumption;

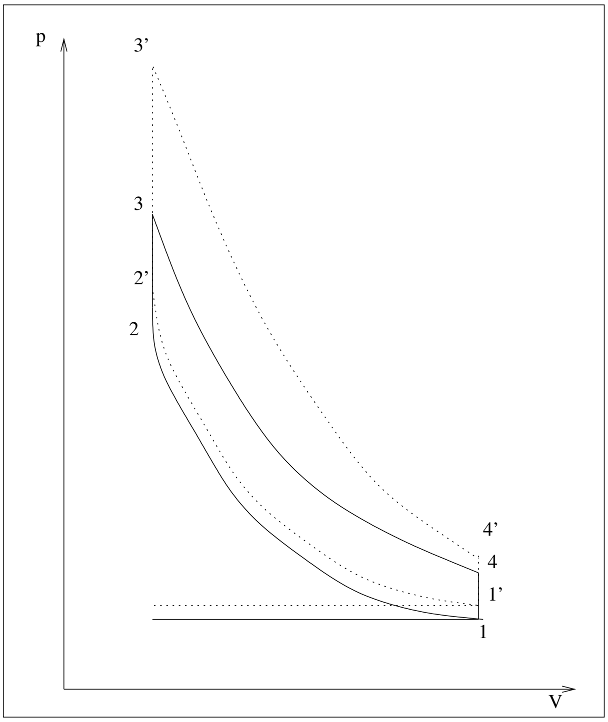


Figure 2.4: SI thermodynamic cycle with and without turbo charging

3. reduces specific emissions due to fuel consumption reduction;
4. the engine become quieter, because temperature has smaller gradient and the turbine reduce pressure waves on the exhaust;
5. especially for aeronautic engines, the motor does not have large power losses if it works in high altitude environment.

Otherwise, there are disadvantages, too:

1. engine must be more resistant, because of larger pressures and forces;
2. in SI engines there is knock danger due to higher pressure in compression phase;
3. turbine needs some times to accelerate, that causes lag phenomena during engine regime variation.

In the last years concepts were developed from most important automotive companies and direct consequent, is downsizing.
As explained, turbo charging increases p_{me} of engines by increasing the power. This concept can also be reconverted to keep constant power reducing engine dimensions.
However, nowadays there is a clear trend in engines construction, some companies prefer to use turbo charging others prefer keep normal charging. Although the theory says turbo charging has more advantages, some problems must be considered on downsizing.
Cylinder dimension reduction means also less space for injection and combustion processes. In the first case issues are fuel's particle reaching more easily wall or piston and

this interaction could impinge on the wall. Impinged mass will not vaporize and never mix with the air, so does not produce energy due to burning while producing particles emission.
Combustion process has similar issue, which are caused by flame-wall interaction producing dissipation and flame extinguishment. It is shown flames More often flame blows out easier combustion results incomplete, reducing produced heat and increasing pollutant.
All of those factors create an incertitude on the real advantages to use. So engine companies have not a clear trend on develop this strategies all over the world. Maybe in next years, after improvement of simulation analysis, ti will be possible to study those phenomena and determine which kind of strategy is the best.

2.6 Emissions

Pollutant control and reduction is one of the biggest challenge that an engine company has to face during development. Since '90 government of the most important industrialized countries introduce every couple of years new rules for engines emissions. For example in Europe after 1992, when emission standard Euro 1 was introduced, every four or five years a new standard is imposing new emissions limits. Meaning engine companies must develop products comply the law, after that other aspects could be considered (price, power . . .).
The following subchapters will be shortly described the major pollutants production inside internal combustion engine.

2.6.1 Carbon Dioxide

Although this gas is the natural results at the combustion, massive production cause greenhouse effect.
This kind of pollutant will be reduce by reducing fuel consumption and complete combustion.

2.6.2 Carbon Monoxide

This gas is results of a partial combustion of fuel. It is a poisonous gas, because it forms a better binding to hemoglobin than oxygen. CO formation is very quick in the flame zone and the major concentration has maximum value on the flame front. CO concentration on the exhaust depends on the fuel/air equivalence ratio, because only when CO is present, CO_2 can be formed. The principal is:

$$CO + OH \rightarrow CO_2 + H \tag{2.14}$$

this is slower than the CO formation reaction and needs intermediate species. So if the fuel ratio is too high this reaction is limited and on exhaust there will be high CO

concentration instead CO_2.

2.6.3 Unburned Hydrocarbon

Those gas are results of partial combustion, the largest source of those are located where the flame extinguishes like cold walls, crevices or other parts where the flame has problem to propagate. Of course, also the fuel/air mixing has strong influence, largest the amount of fuel larger amount of unburned hydrocarbon. Same problem appears if the spark starts too late or engine speed is too high and the flame has no time to propagate through the whole volume.

2.6.4 Nitrogen Oxides

Nitrogen oxides NO_x is a collection of molecules, nitrogen monoxide NO and dioxide NO_2, both are very dangerous for humans in particular for respiratory system. Furthermore, ground level ozone is mainly formed by emitted NO_x.
Principal component (ca. 98%) inside engines emissions is nitrogen monoxide, its formation reactions are:

$$O + N_2 \leftrightarrow NO + N \tag{2.15}$$

$$N + O_2 \leftrightarrow NO + O \tag{2.16}$$

$$N + OH \leftrightarrow NO + H \tag{2.17}$$

this mechanism was proposed by Zeldovich [41] and supposes that nitrogen and oxygen molecules disassociate due to high temperatures. So NO production happens during first phase of combustion and is strong dependent on the maximum temperature reached and on fuel/air mix fraction.
Engine parameters influencing those variables are: compression rate, power, regime, spark timing and EGR (Exhaust gas recirculation) percentage in combustion chamber.

Figure 2.5 shows trend of different pollutants respect to air/fuel mixture. α_0 is the stoichiometric concentration. On the right side there is lean burning zone, on the left side the rich zone. As explained before nitrogen oxides production has strong dependency in temperature. So the function has maximum near stoichiometry where the fuel is enough to burn with high temperatures but not too high missing oxygen and reacting with all the fuel.
Fuel consumption is the optimal parameter to show carbon dioxide function, because as already said they are directly connected. The minimum is located for lightly lean mixture. If mixture becomes richer there is more fuel, so it is logical consumption increment, while increasing formation of CO instead of CO_2. On the other side if mixture is too lean there is more extinguish possibilities, that brings to unoptimized combustion and increment of fuel consumption, because of unburned hydrogen.
Same motivations as above are for unburned hydrocarbons that follow fuel consumption and CO_2.
Very simple is reducing carbon monoxide that must find enough oxygen to react further,

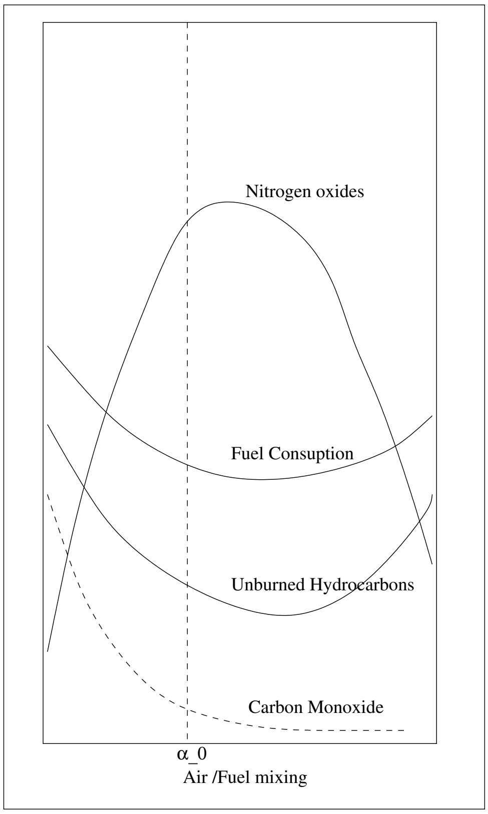


Figure 2.5: Qualitative plot of principal pollutant production function with air/fuel mixture [9]

lean the mixture less production.

Chapter 3

Flow and Combustion Physics in IC engines

In last chapter it was explained IC engines are very complex system incorporating a lot of physical phenomena. Therefore, they are really complex to study. This chapter gives a briefly introduction of the basic concepts including IC engine simulation. In particular, standard equation for physics are presented including flow motion, turbulence and combustion. Afterward it will be explained how to use the equations to analyze IC engine in an instationary computer simulation.

3.1 Flow physics

Fluid motion laws are based on the concept that nothing could be created nor destroyed, hence equations are called conservation equation and are based on very simple principles.

3.1.1 Continuity equation

Analyzing an open system, because mass cannot create nor destroyed, mass entering in:

$$\dot{m} = \sum_i \dot{m}_i \tag{3.1}$$

whereas $\dot{m}$ is the mass inside the system and $\dot{m}_i$ are fluxes entering and leaving.
Considering time variation of the equation 3.1, system mass gradient can also be written with respect to density, and the fluxes in according with Gauss are written as diffusive terms [30].

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0 \tag{3.2}$$

whereas u_i are the velocity components in x_i directions that transport mass inside the system, if ρ is constant (incompressible fluid theory), the equation is very simple. Otherwise, compressible fluid are common in engines, so conservation equation cannot be simplified.

3.1.2 Momentum equation

This equation is based on, Newton's second law (3.3). Fluid acceleration is the result of the applied forces on its. Forces can be applied on volume or surface. Gravity as typical volume force could be written as potential variation (3.4). Stress due to viscosity is a typical surface force and could be written with variation of stress tensor (3.5).

$$mF_v + F_s = ma \tag{3.3}$$

$$F_v = -\nabla\Psi = g \tag{3.4}$$

$$F_s = \nabla\tau_{ij} \tag{3.5}$$

In differential form of Newton's second law leads to [30]:

$$\rho\frac{DU}{Dt} = \frac{\partial\tau_{ij}}{\partial x_i} - \rho g \tag{3.6}$$

While all fluids considered in this work are Newtonian the equation can be specialized for those with Stokes hypothesis [38]:

$$\tau_{ij} = \rho\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\rho\nu\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{3.7}$$

τ_{ij} is symmetric stress tensor for deformation, ν the kinematic viscosity and δ_{ij} Kronecker delta.
Combining equations (3.6) and (3.8) gives Navier Stokes equations (3.8) and fully describe fluid motion [20]:

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\rho\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\rho\nu\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] - \frac{\partial p}{\partial x_i} + \rho g_i \tag{3.8}$$

3.1.3 Scalar transport equation

Other equations could be deduced for every conserved scalar. This kind of equation has four standard terms, time derivative, convection term, diffusion term and source term [20]:

$$\frac{\partial\Phi}{\partial t} + \frac{\partial(\Phi u_j)}{\partial x_j} = \frac{\partial D_j(\Phi)}{\partial x_j} + S_\Phi \tag{3.9}$$

A scalar of this kind could be for example mixture fraction of chemical species, which can be created or consumed during chemical reactions. First term represents scalar accumulation, for this example mixture fraction could increase or decrease. Second term represents the convection of the scalar due to velocity u_j in the direction x_j. Third term depends on diffusion property of the scalar $D_j(\Phi)$. Last one is the source term. On contrary of mass the scalar could be created or destroyed. In example for mixture fraction this term represents the creation or consumption of the species due to chemical reaction. Of course for every increment of mass fraction of one species there must be another one that reduces its mass fraction, to satisfy total mass conservation. The results for transport equation for species m results [2]:

$$\frac{\partial \rho_m}{\partial t} + \frac{\partial(\rho_m u_j)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\rho D \frac{\partial}{\partial x_j}\left(\frac{\rho_m}{\rho}\right)\right] + \dot{\rho}_m^c + \dot{\rho}_m^s \tag{3.10}$$

where $\dot{\rho}_m^c$ is the source due to chemistry and $\dot{\rho}_m^s$ due to spray. D is single diffusion coefficient, assuming Fick's Law. Defining mass fraction for every species as $Y_m = m_m/m$, W_m the molecular weight of species m and X_m the mole fraction of species m: $X_m = Y_m W/W_m$ the equation could be written also [2]:

$$\frac{\partial \rho Y_m}{\partial t} + \frac{\partial(\rho u_j Y_m)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\rho D_m \frac{W_m}{W}\frac{\partial X_m}{\partial x_j}\right] + \dot{\rho}_m^c + \dot{\rho}_m^s \tag{3.11}$$

3.1.4 Energy equation

Another equation for reacting flows could be written with the second law of thermodynamic for open systems:

$$\dot{E} = \dot{Q}_w - \dot{W} + \sum_j \dot{m}_j h_j \tag{3.12}$$

it shows the energy accumulation in a system is the sum of exchanged heat and work with external environment and the energy flux entering and leaving the system. In equation 3.3.1, $\dot{Q}_w$ is the heat transfer rate exchanged with environmental through system boundaries which is positive if the heat enters into the system, negative if leaving. $\dot{W}$ is the work transfer rate out of the system. If work enter the system is negative otherwise is positive. Last term is the sum of energy fluxes entering and leaving the system, including enthalpy inside this fluxes is not necessary include combustion sources.

Turning this law into fluid dynamic analysis is more complex than previous. In this case the energy will be written with internal energy the rest is written following standard transport equation [2]:

$$\frac{\partial(\rho I)}{\partial t} + \nabla \bullet (\rho \bar{u} I) = -p \nabla \bullet \bar{u} + \sigma : \nabla \bar{u} - \nabla \bullet \bar{J} + \dot{Q}_c + \dot{Q}_s \tag{3.13}$$

whereas first term is energy accumulation, second one is convection term that substitutes energy fluxes. Third term represents the work done with motion. Fourth term is the diffusion term and σ is the viscous stress tensor for Newtonian fluid with Stroke's relation. Last three terms are heat exchange and sources. $\bar{J}$ is the heat flux vector, sum of contribution due to heat conduction and enthalpy diffusion. $\dot{Q}_c$ is heat source due to chemical reaction and $\dot{Q}_c$ is the heat source due to spray injection.

This equation could also be written in the form suggested by Poinsot [29]:

$$\frac{\partial \rho I}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i I) = \frac{\partial}{\partial x_i}\left(\lambda \frac{\partial T}{\partial x_i}\right) - \frac{\partial}{\partial x_i}\left(\rho \sum_{m=i}^{N} h_{s,m} Y_m V_{m,i}\right) + \sigma_{ij}\frac{\partial u_i}{\partial x_j} + \dot{Q} + \rho \sum_{m=1}^{N} Y_m f_{m,i} V_{m,i} \tag{3.14}$$

where σ_{ij} is the combination of stress tensor τ_{ij} and pressure tensor $p\delta_{ij}$, $f_{m,i}$ is the power produced by volume forces on species m and $\dot{Q}$ in this case is combination of combustion and exchanged heat.

3.2 Turbulence

All engineering real flows are turbulent or have at least an instability point after that the flow gets chaotic and difficult to predict.
Every flow could change from laminar into turbulent, when came to Reynolds number for dynamic similarity defined as rate of inertial forces to viscous stress, Osborne Reynolds [33], [34]:

$$Re := \frac{M_{ij}}{\tau_{ij}} = \frac{\rho \bar{u} L}{\mu} \tag{3.15}$$

with Reynolds number is possible to determine the status of a flow just knowing density (ρ), velocity ($\bar{u}$), characteristic geometry length (L) and fluid viscosity (μ). In a pipe one could consider fully developed turbulent flow when $Re > 4000$.
In a turbulent flow interaction between fluid layer are strong, dissipation increases and 3-dimensional chaotic structures start to be created. Due to the random nature of turbulent flow is mostly impossible to predict in every moment with an equation. So turbulent model must be adopted to predict at least statistical fluctuations. One of the most famous and simplest method to do it is an incremented viscosity with a fluctuating term.
Although turbulence has a lot of disadvantages for real and simulated flows, this result very useful in combustion systems because it helps with fuel/air mixing as well as combustion.

3.3 Combustion

This section provides a small introduction to the theory of combustion as relevant for this work. Due to great complexity of the topic its mathematical development began few years ago. First simulation's possibilities came with supercomputer development at the end of '90. Turbulent flame simulation must deal with flow motion, heat exchange, turbulence, species diffusion and chemical reaction. Furthermore, simulation of IC engines must deal also with moving computational domain, two phase flow and ignition.

3.3.1 Reaction kinetics

Important parameter to decide in a combustion simulation is the number of chemical reactions activated. During the simulation those reactions are activated and interact with Navier-Stokes, Energy and species balance equations. Reactions itself are not depending on the flow field. Schematic reaction could be:

$$A + B = C + D \tag{3.16}$$

or written in a standard form for more species:

$$\sum_k a_{kj} x_k = \sum_k b_{kj} x_k \tag{3.17}$$

whereas on m is the total number of species, the left side shows the sum of reactants and on the right side the products. x_m is the mole fraction of species k, a_{kj} and b_{kj} are stoichiometric coefficients of species k in reaction j. The balance is true for all reactions considered.

Much comfortable is defining the reaction with mass reaction rate $\dot{\omega}_k$:

$$\dot{\omega}_k = \sum_j \dot{\omega}_{kj} = W_k \sum_j a_{kj} \Omega_j \tag{3.18}$$

Ω_j is defined has progress variable for reaction j. In respect of mass conservation equation:

$$\sum_k \dot{\omega}_k = \sum_j \left(\Omega_j \sum_k W_k a_{kj} \right) = 0 \tag{3.19}$$

the sum of mass reaction rate for all the species is zero.

Progress rate Ω_j must be defined in according to chemical reaction, which includes forward reactions and backward reactions leading to an equilibrium state for the whole reaction mechanism[2]:

$$\Omega_j = K_{fj} \prod_k \frac{\rho_k}{W_k}^{a_{kj}} - K_{rj} \prod_k \frac{\rho_k}{W_k}^{b_{kj}} \tag{3.20}$$

where K_{fj} and K_{rj} are forward and revers rate constant. Evaluated from the empirical Arrhenius law [21]:

$$K_{fj} = A_{fj} T^{\beta_j} \exp\left(-\frac{E_j}{RT} \right) = A_{fj} T^{\beta_j} \exp\left(-\frac{T_{aj}}{T} \right) \tag{3.21}$$

written in dependency of activation energy E_j or activation temperature T_{aj}, both indicating the minimum level of energy to activate the reaction. The equation could be written for both reaction's directions. A_{fj} is called preexponential factor, together with temperature exponent β_j and activation temperature or energy, are defined from complex kinetic scheme proposed. It is really important when set of reaction is defined that all those factors are available and documented, otherwise the chemical kinetic can not solved for the whole simulation.

If progress rates for every reaction are calculated, chemical sources and heat release will be included in flow equations.

The chemical sources for every species is calculated and included in the species continuity equation, relation is given by:

$$\dot{\rho}^c_m = W_m \sum_r (b_{mr} - a_{mr}) \dot{\omega}_r \tag{3.22}$$

chemical heat release term to include as source in energy equation () is given by:

$$\dot{Q}_c = \sum_r \left(\sum_m (a_{mr} - b_{mr}) \Delta h^0_{fm} \right) \dot{\omega}_r \tag{3.23}$$

3.3.2 Flame classification

Although in real applications a lot of different combustion cases exist, for investigation is useful to distinguish between two limiting cases: premixed flame and diffusion flames.
Premixed combustion happens when the mixture fraction is homogeneous in the studied domain. In premixed combustion air/fuel is homogeneously distributed in studied domain. The flame propagates from ignition point through the combustion chamber, until mixture reaches equilibrium state. Typical example for this kind of combustion in engine are gasoline engines. Diffusion combustion is the exact contrary of the above case, fuel and oxidizer are separated and flame propagate during the mixing. For those cases ignition is not needed. The combustion starts in different points and propagates where mixing and flammability limit are reached. Diesel engines are the standard example of diffusion flame combustion.

3.3.2.1 Premixed flames

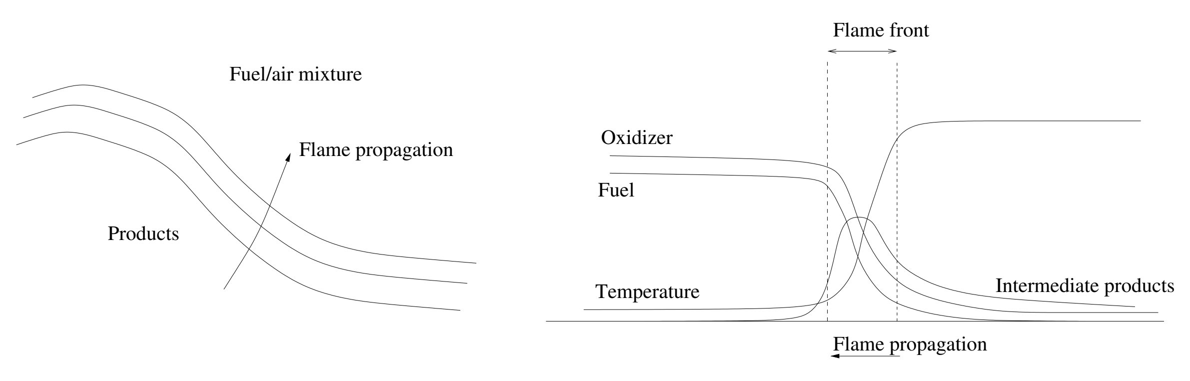


Figure 3.1: Schematic representation of 1D premixed flame [16]

Simple analysis of flame can be done considering the propagation of one dimensional laminar flame.
Fuel and air are perfectly premixed but initial temperature are far below from flammability point. Reaching those conditions external factor must be introduced, usually is an energy source (for IC engines is the spark plug). Once activation energy is reached the flame propagates without any kind of external energy source.
Typical distributions of species and temperature is shown in Figure 3.1. On the left side, where the unburned gas is located, oxidizer and fuel mass fractions are at their maximum values, temperature is very low. The central zone on the graphic is the reaction layer or flame front, at this state the gradient are high and the reactions have the maximal velocity. Fuel and oxidizer disappear, intermediate and products are created. Principal difference between products and intermediate species is related to the stability of the molecules. Intermediate species are instable whereas CO_2 and H_2O are formed as stable

products. Intermediate species have their maximum inside the flame front, after that they are consumed to participate in stable product formation.
For premixed flame is possible to define flame thickness and flame speed, which are two important parameter to validate chemistry mechanism or new implemented models.
Additionally, flame thickness is important because the grid dimension for calculation is based on it, is necessary to calculate this in advance. There are more method to define this parameter and some of them require first computation of 1D flame. Based on Poinsot [29] simplest and most useful definition are three equation.

$$\delta = \frac{\lambda_1}{\rho_1 C_p s_L} = \frac{D_{th}^1}{s_L} \tag{3.24}$$

where ρ_1 is the density and λ_1 is the thermal conductivity, which are evaluated on the fresh gas. This is called "diffusive" thickness, and can be calculated only if laminar flame speed S_L is known. Usually, this formulation is too inaccurate to be used for mesh determination.
The "total" thickness is based on the temperature profile, results of simulation:

$$\delta_L^0 = \frac{T_2 - T_1}{max\left(\left|\frac{\partial T}{\partial x}\right|\right)} \tag{3.25}$$

"Total thickness" is constructed after computation and is defined with distance over reduced temperature Θ (3.26), which changes from 0.01 to 0.99, see Figure 3.2

$$\Theta = \frac{T - T_1}{T - T_2} \tag{3.26}$$

Other important parameter for model and chemistry validation is the flame velocity. Unfortunately this speed is not easy to calculate and one could find a lot of different definitions. Hence, this work does not present the whole discussion but gives some simple definitions.
Poinsot [29] defines three different flame speed:

Absolute speed whereas the flame is moving;

Displacement flame speed relative to flow reference frame;

Consumption speed at which reactants are consumed.

Figure 3.3 shows the notations for definitions. s_d is flame displacement, u the flow velocity in the point of interest, n is the normal direction in that point and w is the absolute flame speed. Defining the flame front as isosurface of reduced temperature Θ, flame normal direction is in every point defined as:

$$\bar{n} = -\frac{\nabla\Theta}{|\nabla\Theta|} \tag{3.27}$$

Motion law of the point from time t to $t + dt$ can be written as:

$$\frac{\partial\Theta}{\partial t} + \bar{w} \bullet \nabla\Theta = 0 \tag{3.28}$$

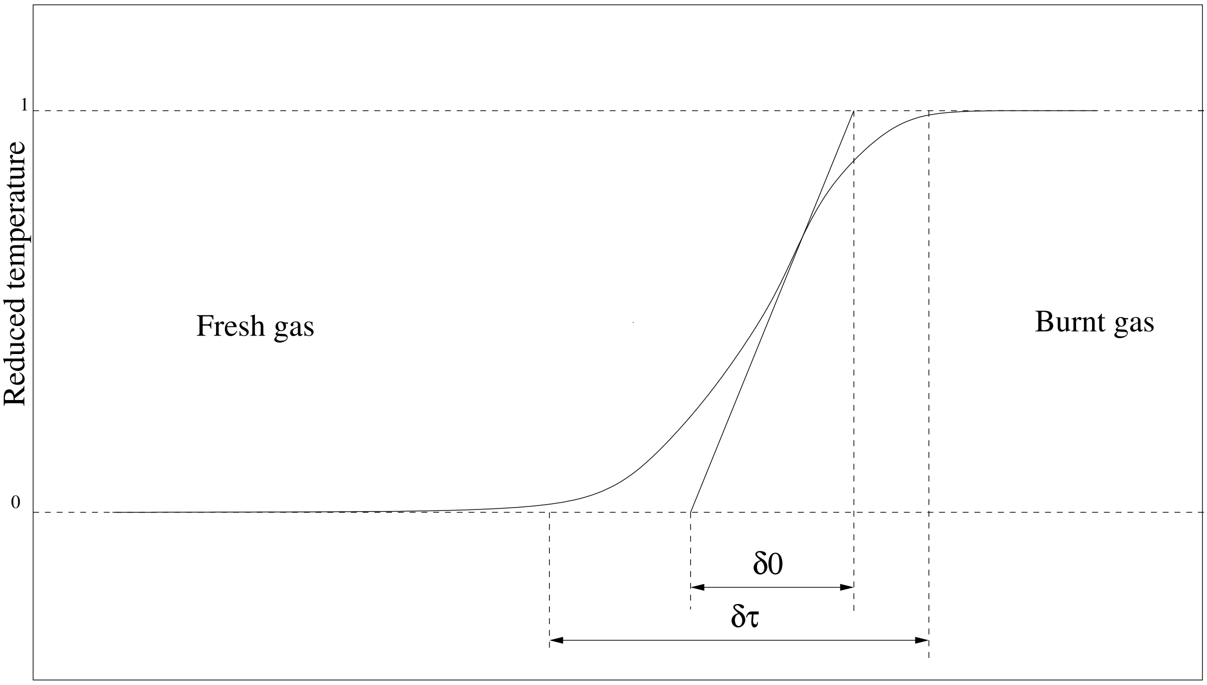


Figure 3.2: Graphic of thickness definitions

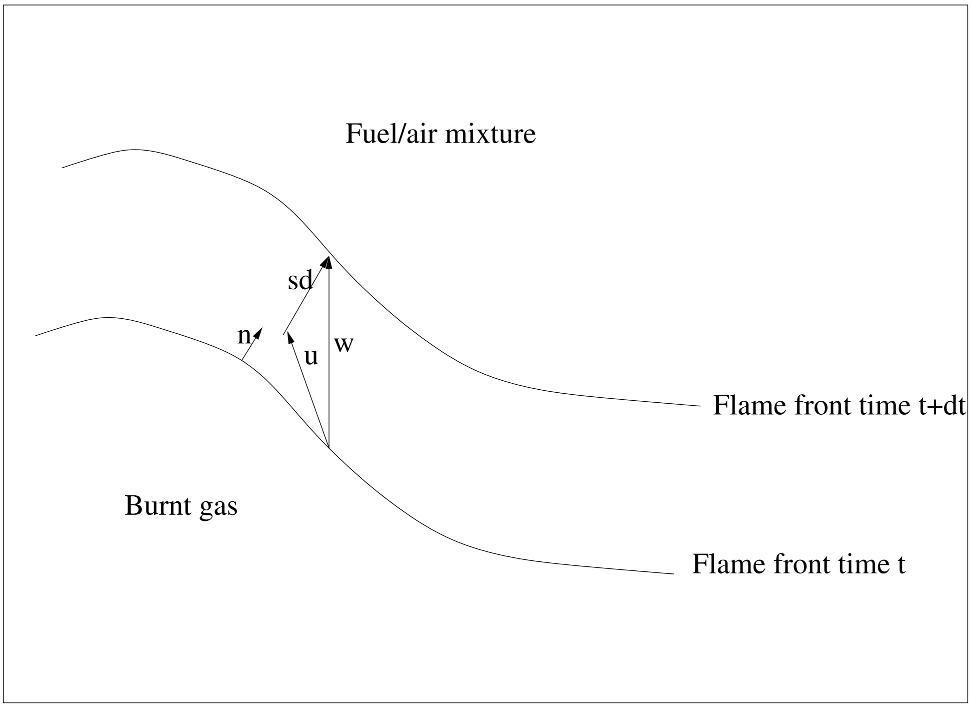


Figure 3.3: Notations for flame speed definition

Defining local flame velocity as normal component of $\bar{w}$:

$$s_a = \bar{w} \bullet \bar{n} = \frac{1}{|\nabla\Theta|}\frac{\partial\Theta}{\partial t} \tag{3.29}$$

Displacement speed is just the difference between previous and flow velocity:

$$s_a = (\bar{w} - \bar{u}) \bullet \bar{n} = s_a - \bar{u} \bullet \bar{n} \tag{3.30}$$

The last definition is not a speed. It is based on reactant consumption:

$$s_c = \frac{1}{\rho_1 Y_F^1} \int_{-\infty}^{+\infty} \dot{\omega}_F dn \tag{3.31}$$

Important difference between these speeds is, that s_d and s_a are local quantities depending on flame position and flame/turbulence interaction, the last one is a global definition calculated as average and not considering fluctuations.
The laminar flame velocity S_L^0 is the reference dimension used in all combustion studies to validate simulations, definition for this parameter is:

$$s_L = \frac{u - u_1}{\frac{\rho_1}{\rho} - 1} \tag{3.32}$$

In case of 1D flame propagation u and u_1 are flow velocity in outlet and inlet or in points far away from the flame front.

3.3.2.2 Diffusion flames

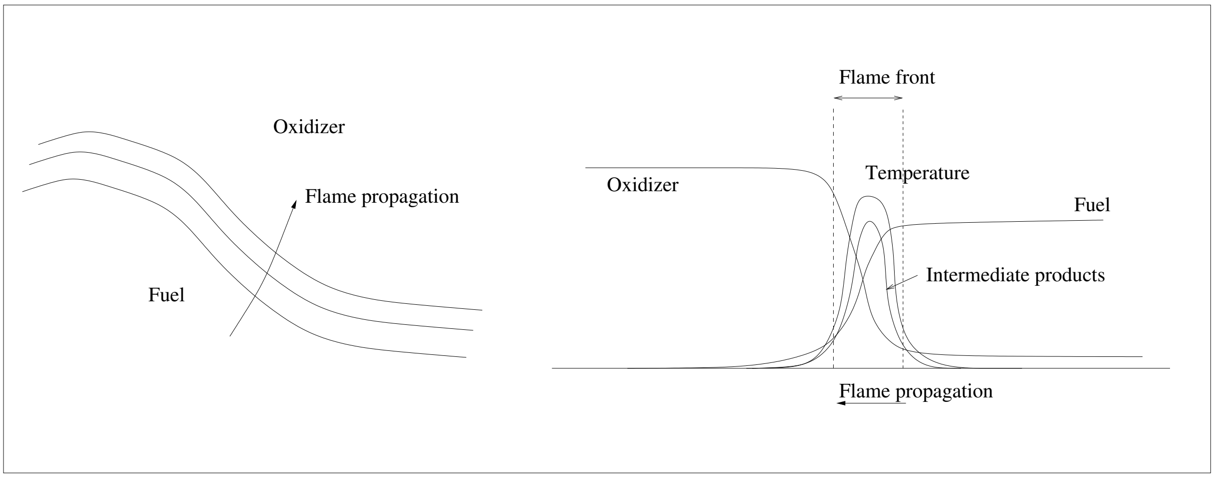


Figure 3.4: Schematic representation of 1D diffusion flame [16]

One dimensional description of diffusional flames has some similarities with the premixed flames. However, the characteristic of those flames are really different.
Figure 3.4 shows species and temperature distributions for a diffusion flame. In diffusion flame the major problem is the mixing, is determining burning condition. Far away from the reaction layer the gas is too rich or too lean to react. So only if the mixing is accurate done the combustion takes place. In this case the gas need to move to mix together. Fuel or oxidizer must be pushed on the direction of each other to react. For this kind of flame is impossible to define a flame velocity as premixed flames because the flame is not

properly propagating.
For industrial applications those flames are safer as premixed flame, because the gases are always separated. So, in case of electric spark or some sort of energy source before the combustion chamber is not dangerous, for a premixed flame this will be a real problem because the reactions is able to start.

Chapter 4

Modeling turbulent reactive flows

In this chapter the most important modeling strategies of turbulent flows are presented with a slightly deeper description of model used in this work.
The layout of the chapter is structured with the complexity of the simulation, starting from the most expensive concerning computational time DNS (direct numerical simulation) to the less expensive LES (large eddy simulation) and all the models used for this work are presented. RANS (reynolds average navier-stokes) isn't interesting for this work, so won't be explained here.

4.1 Energy cascade

Before explaining the different simulation methods, the concept of energy cascade must be introduced. This topic is really important because all the method are based on this.
Considering a fully turbulent flow at high Reynolds number, with characteristic velocity U and lengthscale L. Turbulence inside this flow is composed of eddies of different characteristic sizes. Every eddy has size l, characteristic velocity $u(l)$ and timescale $\tau(l)$=l/u(l). Biggest eddies have lengthscale l_0 comparable to L.
Richardson's notion is that the large eddies are unstable and break up, transferring their energy to smaller eddies and creating the energy cascade phenomena.
Phenomena is called cascade because energy transfer happens at every lengthscale until Reynolds number $Re(l) = u(l)l/\mu$ is small enough to stabilize eddy motion, at this point kinetic energy is dissipated in thermal energy by molecular viscosity.
Really important for simulation is the Kolmogorov's hypothesis that says [30]:

> "*At sufficiently high Reynolds number, the small-scale turbulent motion ($l \ll l_0$) are statistically isotropic*"

The lengthscale where this happens is defined as $l_{EI} = \frac{1}{6}l_0$. Eddies with $l > l_{EI}$ are large and anisotropic this range is called energy-containing, on the other hand small eddies with $l < l_{EI}$ are isotropic and the range is called universal equilibrium.
The universal equilibrium range can be separated in two different subranges using Kolmogorov's similarity hypothesis, the separation lengthscale is called l_{DI}. The two subranges are called inertial subrange and dissipation subrange.
Inertial subrange limits can be written as $l_{EI} > l > l_{DI}$, in this motion are determined by inertial effects and viscous effects are negligible. Dissipation range is the lowest one and includes the lengthscale below l_{EI}, here the major effect is the viscous one and is

responsible for energy dissipation.
Figure 4.1 shows how turbulent kinetic energy is dissipated when the wavenumber $k = 2\pi/l$ is increasing. The last part of the curve is very sloped because the viscous effect are very strong.

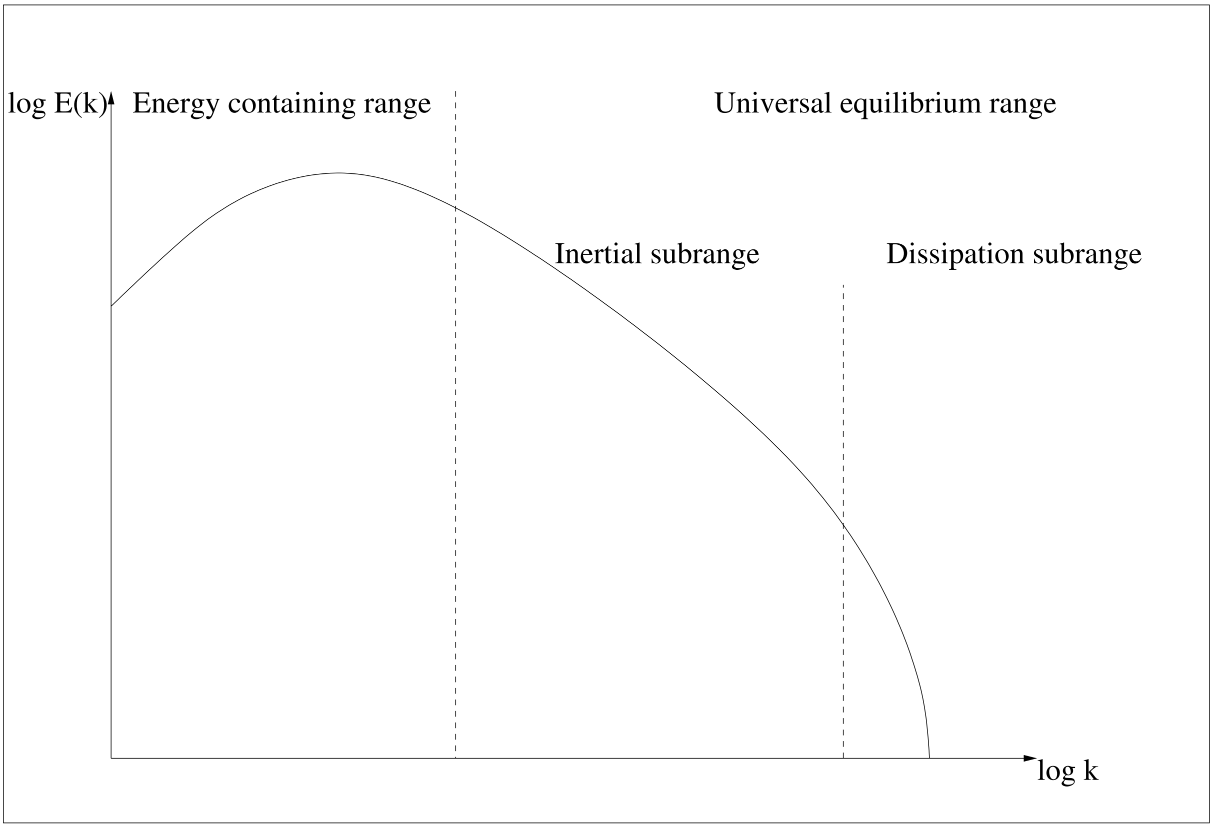


Figure 4.1: Typical distribution of turbulent kinetic energy with wavenumber

4.2 Direct numerical simulation

Explaining DNS method is presented. Due to very good accuracy of Navier-Stokes equations describes all sort of flows without modeling, but only for simple flows solvable analytically. However equations don't have an analytical solution and they must be discretized in space and time. Because of the non modeling, scalar and spatial resolution must be chosen enough small to resolve all scales. The computational costs very high because the dimension of the cell must be small enough to resolve also are the structures of dissipative subrange.
Kolmogorov scales are defined as [30]:

$$\eta \equiv \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}} \tag{4.1}$$

$$u_\eta \equiv (\varepsilon\nu)^{\frac{1}{4}} \tag{4.2}$$

$$\tau_\eta \equiv \left(\frac{\nu}{\varepsilon}\right)^{\frac{1}{2}} \tag{4.3}$$

whereas η is the lengthscale, u_η is the velocity and τ_η is the time scale, all defined with dissipation rate ε and kinematic viscosity ν.
Since the dissipation rate can be written as function of mean velocity:

$$\varepsilon \sim \frac{\bar{U}^3}{L} \tag{4.4}$$

can be shown that problem scales are dependent on Reynolds number:

$$\eta \sim \left(\frac{1}{Re}\right)^{\frac{3}{4}} \tag{4.5}$$

$$\tau_\eta \sim \left(\frac{1}{Re}\right)^{\frac{3}{2}} \tag{4.6}$$

Considering now the resolution of all scale it is necessary to include a cell dimension at least 2-3 times smaller than η and this will be true for three directions in the space, the total amount of nodes needed high also for Low Reynolds number problems and is impossible to solve for High Reynolds number analysis.

4.3 Large-eddy simulation

Because the complete absence of modeling is still unreachable for the majority of engineering problems, less expensive method are considered.
Large-eddy simulation in the last years become more used and now on the scientific environment is one of the most developed method. The concept is to resolve the biggest structures and model the small structures for while isotropic assumption is more realistic. Of course the method isn't precise as DNS but the computational costs are limited and affordable for the actual computational power.
While calculated part isn't complete, modeling is still needed. A filter operation is applied on the standard Navier-stokes equation. Resolved component of total velocity $U(x,t)$ will be called filtered velocity $\overline{U}(x,t)$. Difference between those two components will be modeled and called subgrid-scale (sgs) velocity $u'(x,t)$.
The sgs component is calculated by means of the residual stress tensor, that by the results filtering of the Navier-stokes equations.
Filtering operator was first proposed by Smagorinsky [37]. A low pass filtering is used to reduce computational costs of DNS and perform a simulation with larger cells and increased time discretization width.
General filtering operation was introduced by Leonard [23], defined as:

$$\overline{U}(x,t) = \int G(r,x)U(x-r,t)dr \tag{4.7}$$

where the integration contains the entire flow domain and the specified filter function G satisfies the normalization condition.

$$\int G(r,x)dr = 1 \tag{4.8}$$

A typical filter function is the box filter of $\Delta = (\Delta_x \Delta_y \Delta_z)^{\frac{1}{3}}$ dimension, defined as:

$$G(x_j - x_j') = \begin{cases} \prod_{j=1}^{3} 1/\Delta_j & |x_j'| \leq \frac{\Delta_j}{2} \\ 0 & \text{else} \end{cases} \tag{4.9}$$

An other important filter operator is the Favre filtering, useful for fluid-dynamic compressible simulation where the density varies. Favre filter is a density weighted operator without any correlation between density and velocity.

$$\widetilde{\Phi} = \frac{\overline{\rho\Phi}}{\overline{\rho}} \quad \text{with} \quad \Phi = \widetilde{\Phi} + \Phi'' \tag{4.10}$$

in this case Φ'' is still the small structures contribution. But because of the different filtering operator the symbol was changed.

4.3.1 Filtered governing equations

Filtering the standard governing equations allow to resolve the flow with LES method, but with this operation new terms inside equations appear. Those term must be modeled. Following filtered equation will be presented, followed by model possibility.

Continuity equation

The continuity equation is the only equation that has no additional term than the original one due to filtering operator:

$$\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial \overline{(\rho u_i)}}{\partial x_i} = 0 \quad \text{or} \quad \frac{\partial \overline{\rho}}{\partial t} + \frac{\partial (\overline{\rho} \widetilde{u_i})}{\partial x_i} = 0 \tag{4.11}$$

Momentum equation

Same operation will be done for Navier-Stokes equations:

$$\frac{\partial \overline{(\varrho u_i)}}{\partial t} + \frac{\partial \overline{(\varrho u_i u_j)}}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\overline{\varrho \nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)} - \frac{2}{3} \overline{\varrho \nu \frac{\partial u_k}{\partial x_k}} \delta_{ij} \right] - \frac{\partial \overline{p}}{\partial x_i} + \overline{\varrho} g_i \tag{4.12}$$

applying Favre filter:

$$\frac{\partial (\overline{\varrho} \widetilde{u_i})}{\partial t} + \frac{\partial (\overline{\varrho} \widetilde{u_i u_j})}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\overline{\varrho} \left(\nu \widetilde{\frac{\partial u_i}{\partial x_j}} + \nu \widetilde{\frac{\partial u_j}{\partial x_i}} \right) - \frac{2}{3} \overline{\varrho} \nu \widetilde{\frac{\partial u_k}{\partial x_k}} \delta_{ij} \right] - \frac{\partial \overline{p}}{\partial x_i} + \overline{\varrho} g_i \tag{4.13}$$

In this equation there are unknown terms. First is the diffusion term that can be approximated by [20]:

$$\nu \widetilde{\frac{\partial u_i}{\partial x_j}} \approx \widetilde{\nu} \widetilde{\frac{\partial u_i}{\partial x_j}} \tag{4.14}$$

second unknown is the term $\widetilde{u_i u_j}$, that can be considered as the sum of resolved part and modeled part:

$$\widetilde{u_i u_j} = \widetilde{u}_i \widetilde{u}_j + \tau_{ij}^{sgs} \tag{4.15}$$

The new term τ_{ij}^{sgs} also known as subgrid stress is the modeled part of the shear stress due to the turbulent motion. Writing the equation comprehensive of (4.14) and (4.15), the results is similar to the original one but with subgrid stresses and the diffusion term:

$$\frac{\partial(\overline{\varrho}\widetilde{u}_i)}{\partial t} + \frac{\partial(\overline{\varrho}\widetilde{u}_i\widetilde{u}_j)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\overline{\varrho}\widetilde{\nu}\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i}\right) - \frac{2}{3}\overline{\varrho}\widetilde{\nu}\frac{\partial \widetilde{u}_k}{\partial x_k}\delta_{ij} + \overline{\varrho}\tau_{ij}^{sgs}\right] - \frac{\partial \overline{p}}{\partial x_i} + \overline{\varrho} g_i \tag{4.16}$$

Chemical species

Filtered species mass fraction balance equation has also a new term that need to be modeled:

$$\frac{\partial \overline{\rho}\widetilde{Y_m}}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u}_j\widetilde{Y_m})}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\overline{V_{m,i}Y_m} - \overline{\rho}\left(\widetilde{u_i Y_m} - \widetilde{u}_i\widetilde{Y}_k\right)\right] + \overline{\dot{\rho}_m^c} + \overline{\dot{\rho}_m^s} \tag{4.17}$$

whereas $V_{m,i}$ is the i-component of the diffusion velocity V_k. In this equation the term $\left(\widetilde{u_i Y_k} - \widetilde{u}_i\widetilde{Y}_k\right)$, from Poinsot [29] called unresolved species fluxes because is part of diffusion term, is the modeled component of species diffusion. Using Fick law the filtered diffusion term can be also reconverted in:

$$\overline{V_{m,i}Y_m} = \overline{\rho D \frac{\partial Y_m}{\partial x_j}} \tag{4.18}$$

where D is the molecular diffusivity.

Energy equation

Starting from the presented form of the energy equation, the filtering operation leads to:

$$\frac{\partial \overline{\rho}\widetilde{I}}{\partial t} + \frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{u}_i\widetilde{I}) = \frac{\partial}{\partial x_i}\left(\overline{\lambda\frac{\partial T}{\partial x_i}} - \overline{\rho}(\widetilde{u_i I} - \widetilde{u}_i\widetilde{I})\right) - \frac{\partial}{\partial x_i}\left(\overline{\rho\sum_{m=i}^{N} h_{s,m}Y_m V_{m,i}}\right) + \overline{\sigma_{ij}\frac{\partial u_i}{\partial x_j}} + \overline{\dot{Q}} \tag{4.19}$$

In this case the term $(\widetilde{u_i I} - \widetilde{u}_I\widetilde{I})$, here called internal energy flux, must be modeled.

4.3.2 Terms modeling

In this section the principal methods of previously explained terms that has to be modeled will be explained.

Smagorinsky model [37]

The Smagorinsky subgrid-scale model is the simplest LES model for concept and also for computational request, therefor is often used.
This model is based on the Boussinesq assumption for turbulent Reynolds stresses:

$$\tau_{ij} - \frac{\delta_{ij}}{3}\tau_{kk} = -\nu_t \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} \right) = -2\nu_t \overline{S}_{ij} \tag{4.20}$$

where ν_t is called subgrid scale viscosity, while is added to the real viscosity and is calculated as:

$$\nu_t = C_s^2 \Delta^2 |\overline{S}| = C_s^2 \Delta^2 (2\overline{S}_{ij}\overline{S}_{ij})^{1/2} \tag{4.21}$$

C_s is a model constant estimated $0.1 \div 0.2$, Δ is the grid size, assumed of the order of the integral length scale and $\overline{S_{ij}}$ are the components of the resolved shear stress tensor $\overline{S}$. With this kind of model we are assuming an isotropic contribution of the sgs turbulence due to a bigger viscosity of the fluid. The problem of Smagorinsky model is the very high dissipation, imposed from the constant C_s, but other studies demonstrate that close to the wall the constant needs to be reduced.

Germano dynamic model

This model was presented by Germano et al. [13] and Lilly [24] and its objective is to introduce a dynamic C_s variable into Smagorinsky model, to take into account the small scale dissipation.
The basis idea is to use the standard Smagorinsky with a better quality introducing a function to calculate C_s in every discretized point and every time step.
First of all Germano defines besides standard LES filter a second one, called test filter:

$$\overline{f}(x) = \int f(x')\overline{G}(x, x')dx' \tag{4.22}$$

$$\widehat{f}(x) = \int f(x')\widehat{G}(x, x')dx' \tag{4.23}$$

meanwhile the dimension of the first filter is defined as the grid dimension $\overline{\Delta}$, the test filter dimension is larger than the previous one and is defined as $\widehat{\overline{\Delta}}$.
New applied filter function results $\widehat{\overline{G}} = \widehat{G}\overline{G}$.
Already defined subgrid scale stress tensor term:

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j \tag{4.24}$$

applying the new test filter results is:

$$\mathcal{T}_{ij} = \widehat{\overline{u_i u_j}} - \widehat{\bar{u}}_i \widehat{\bar{u}}_j \tag{4.25}$$

Considering that a part of the turbulent stress is resolved using filtered equations, $\mathcal{L}_{ij}$ is defined as:

$$\mathcal{L}_{ij} = \widehat{\bar{u}_i \bar{u}_j} - \widehat{\bar{u}}_i \widehat{\bar{u}}_j \tag{4.26}$$

which represents the contribution of Reynolds stresses by small part of resolved scale. Algebraic relation between equation 4.24, 4.25 and 4.26 is given by:

$$\mathcal{L}_{ij} = \mathcal{T}_{ij} - \widehat{\tau}_{ij} \tag{4.27}$$

Starting from this form an adequate Smagorinsky coefficient can be calculate. Assuming sub grid stress model for both filtering can be written in the same form presented for Smagorinsky model:

$$\tau_{ij} - \frac{\delta_{ij}}{3}\tau_{kk} \approx -2C\overline{\Delta}^2|\overline{S}|\overline{S}_{ij} \tag{4.28}$$

$$\mathcal{T}_{ij} - \frac{\delta_{ij}}{3}\mathcal{T}_{kk} \approx -2C\widehat{\overline{\Delta}}^2|\widehat{\overline{S}}|\widehat{\overline{S}}_{ij} \tag{4.29}$$

where $C = C_s^2$ is the Germano constant:

$$\widehat{\overline{S}}_{ij} = \frac{1}{2}\left(\frac{\partial\widehat{\overline{u}}_i}{\partial x_j} + \frac{\partial\widehat{\overline{u}}_j}{\partial x_i}\right) \quad |\widehat{\overline{S}}| = \sqrt{2\widehat{\overline{S}}_{mn}\widehat{\overline{S}}_{mn}} \tag{4.30}$$

Substituting gives the Germano identity:

$$\mathcal{L}_{ij} - \frac{\delta_{ij}}{3}\mathcal{L}_{kk} = 2C\left(\widehat{\overline{\Delta}^2|\overline{S}|\overline{S}_{ij}} - \overline{\Delta}^2|\widehat{\overline{S}}|\widehat{\overline{S}}_{ij}\right) \tag{4.31}$$

The model is very efficient for a lot of cases although the presented version is reliable for incompressible fluids. Compressible version was presented by Moin et al. [26]
One limitation of this model is that some quantity on C equation can become zero, which would make C indeterminate or ill-conditioned. For that reason in computation C values are limited between zero and maximum value, that is chosen as 0.009 that correspond to Smagorinsky constant $C_s = 0.3$.

Chemical reaction rate modeling

Chemical reaction rate is a very important parameter that needs to be modeled in reacting flow simulation.
Already presented in previous chapter 3, Arrhenius law can also be used in this case, but considering the filtered components:

$$\overline{\dot{\omega}}_F = A_1\overline{\rho}^2\widetilde{Y}_F\widetilde{Y}_O\widetilde{T}^{\beta_1}\exp\left(-\frac{T_A}{\widetilde{T}}\right) \tag{4.32}$$

This expression implicitly assumes that simulation time discretization τ_s is sufficient smaller than chemical time scales τ_c, which is often a good assumption because the Courant number or turbulent time scale impose much smaller time step on the simulation than τ_c. An other issue in LES for premixed flames is the thickness of the flame which is about $\delta_L^0 = 0.1$mm, grid dimension is generally larger that this. Therefore the flame is usually not resolved by the grid of the computational domain. Without a sufficient discretization happen that the progress variable can increase from zero (no reaction) to one (reaction completed) inbetween two neighbor cells. Therefor the flame can be tracked as thick as one cell, or without a thickness. Both options don't allow a proper interaction between flame and turbulence. Nearby computational issue there is also the fact that the most important contribution for turbulent combustion is the one from subgrid scale, so direct Arrhenius law application without any other model results impossible for LES.

Artificially thickened flame model

Basic idea of this model is to modify the thickening of the flame to fit the flame front into LES grid so Arrhenius law can be used without disadvantages. Originally the model was presented by Butler and O'Rourke [10] than improved by Colin et al. [8] and Charlette et al. [6].
Simple flame theory defines flame front thickness δ_l^0 and velocity s_l^0 as:

$$\delta_l^0 \propto \frac{D}{s_l^0} \quad \text{and} \quad s_l^0 \propto \sqrt{D\overline{W}} \tag{4.33}$$

where D is the species diffusion factor and $\overline{W}$ is the mean reaction rate.
Increasing flame thickness is done by modifying thermal and molecular diffusion factors, generally included in non dimensional Prandtl and Schmidt numbers:

$$Pr = \frac{\nu}{D} = \frac{\text{viscous diffusion rate}}{\text{molecular diffusion rate}} \tag{4.34}$$

$$Sc = \frac{\nu}{\alpha} = \frac{\text{viscous diffusion rate}}{\text{thermal diffusion rate}} \tag{4.35}$$

where ν is the kinematic viscosity, D the molecular diffusion, $\alpha = k/(\rho c_p)$ thermal diffusivity and k thermal conductivity.
Furthermore reaction rate must be reduced with the same factor used for diffusivity.
Writing now chemical species filtered equation without thickening is:

$$\frac{\partial \bar{\rho}\widetilde{Y_m}}{\partial t} + \frac{\partial(\bar{\rho}\widetilde{u_j}\widetilde{Y_m})}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\bar{\rho}\left(\widetilde{u_i Y_m} - \widetilde{u_i}\widetilde{Y_k}\right)\right] + \frac{\partial}{\partial x_j}\overline{\left(\rho D\frac{\partial Y_m}{\partial x_j}\right)} + \overline{\dot{\rho}_m^c} + \overline{\dot{\rho}_m^s} \tag{4.36}$$

applying thickening as expressed above:

$$\frac{\partial \bar{\rho}\widetilde{Y_m}}{\partial t} + \frac{\partial(\bar{\rho}\widetilde{u_j}\widetilde{Y_m})}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\bar{\rho}\left(\widetilde{u_i Y_m} - \widetilde{u_i}\widetilde{Y_k}\right)\right] + \frac{\partial}{\partial x_j}\overline{\left(\rho F D\frac{\partial Y_m}{\partial x_j}\right)} + \frac{\overline{\dot{\rho}_m^c}}{F} + \overline{\dot{\rho}_m^s} \tag{4.37}$$

Unfortunately the increased flame thickness also cause a modification in the turbulence-flame interaction because the non dimensional Damkoehler number is changed:

$$Da = \frac{\tau_t}{\tau_c} = \frac{l_t s_l^0}{u'\delta_l^0} = \frac{\text{turbulent time scale}}{\text{chemical time scale}} \tag{4.38}$$

Is demonstrated that a reduction of l_t/s_l^0 makes the flame less sensitive to the turbulence. Important phenomena like flame wrinkling are neglected therefore other modifications must to be implemented.

In the last years different models were proposed to define a proper flame wrinkling. In this work two are analyzed, presented by Colin et al. [8] and Charlette et al. [6].
Both are based on a definition of a wrinkling factor as rate of turbulent flame speed and laminar flame speed:

$$\frac{S_T}{s_l^0} = \Xi \tag{4.39}$$

Colin defines wrinkling factor after analysis of filtered flame density equation as:

$$\Xi\left(\frac{\Delta}{\delta_l^0}, \frac{u'_\Delta}{s_l^0}, Re_t\right) = 1 + \alpha \frac{3}{2} \frac{c_{ms}}{ln(2)} \left(\frac{\Delta}{\delta_l^1}\right)^{\frac{1}{3}} \frac{u'_\Delta}{s_l^0} \left[\left(\frac{\Delta}{\delta_l^c}\right)^{\frac{2}{3}} - 1\right] \Gamma\left(\frac{\Delta}{\delta_l^0}, \frac{u'_\Delta}{s_l^0}\right) \tag{4.40}$$

where u'_{Δ_e} is the subgrid scale turbulent velocity and α is a model constant.
Using Damkoehler theory α can be estimated as:

$$\alpha = \beta \frac{2ln(2)}{3c_{ms}\left[Re^{0.5} - 1\right]} \tag{4.41}$$

where β is a model constant of the order of unity, $c_{ms} = 0.28$ is a model constant calculated by Yeung et al. [19]
The function Γ was fitted as:

$$\Gamma\left(\frac{\Delta}{\delta_l^0}, \frac{u'_\Delta}{s_l^0}\right) = 0.75 \exp\left[-\frac{1.2}{(u'_\Delta/s_l^0)^{0.3}}\right] \left(\frac{\Delta}{\delta_l^0}\right)^{\frac{2}{3}} \tag{4.42}$$

On this approach the factor Ξ requires three different parameters but two different problems arise, the first one is the evaluation of subgrid scale velocity: that can be done with Smagorinsky model formulation, applying an operator on Taylor expansion series, using Laplacian of the resolved vorticity or from calibration of isotropic turbulence constant. Second one is the evaluation of Reynolds number, that is really difficult because it changes every point and time of the simulation.
Because of this two issues Charlette's approach was used in this work. This approach suggests a formulation for Ξ like:

$$\Xi_\Delta\left(\frac{\Delta}{\delta_l^0}, \frac{u'_\Delta}{s_l^0}, Re_\Delta\right) = \left\{1 + \min\left[\frac{\Delta}{\delta_l^0}, \Gamma \frac{u'_\Delta}{s_l^0}\right]\right\}^\beta \tag{4.43}$$

where the function Γ is fitted from various calculations:

$$\Gamma_\Delta = \left[\left(\left(f_u^{-a} + f_\Delta^{-a}\right)^{-1/a}\right)^{-b} + f_{Re}^{-b}\right]^{-1/b} \tag{4.44}$$

where:

$$f_u = 4\left(\frac{27C_k}{110}\right)^{1/2} \left(\frac{18C_k}{55}\right) \left(\frac{u'_\Delta}{s_l^0}\right)^2 \tag{4.45}$$

$$f_\Delta = \left[\frac{27C_k \pi^{4/3}}{110} \left(\left(\frac{\Delta}{\delta_l^0}\right) - 1\right)\right]^{1/2} \tag{4.46}$$

$$f_{Re} = \left[\frac{9}{55}exp\left(-\frac{3}{2}C_k\pi^{4/3}Re_\Delta^{-1}\right)\right]^{1/2} Re_\Delta^{1/2} \tag{4.47}$$

$$a = 0.60 + 0.20\exp\left[-0.1\frac{u'_\Delta}{s_l^0}\right] - 0.20\exp\left[-0.01\frac{\Delta}{\delta_l^0}\right] \tag{4.48}$$

$$b = 1.4 \tag{4.49}$$

In this case the subgrid scale Reynolds number is calculated as:

$$Re_\Delta = 4\frac{\Delta}{\delta_l^0}\frac{u'_\Delta}{s_l^0} \tag{4.50}$$

In this approach there are only two independent parameter instead of three in the previous one equation 4.40.
To estimate subgrid scale velocity u'_Δ in this case the Laplacian of the resolved vorticity is used:

$$u'_\Delta = c_2\Delta^3|\nabla\times\left(\nabla^2(\widetilde{u})\right)| \tag{4.51}$$

where $\widetilde{u}$ is the resolved velocity in the grid size Δ and $c_2 = 2.0$ is a model constant.

Dynamically thickened flame model

Analyzing the physical phenomena modified by ATF is really easy to understand that with modification of molecular and thermal diffusion parameters not only flame propagation is modified, but also diffusion processes are modified. Therefor model modification was presented by Legier et al. [22] called Dynamically thickened flame.
This model take into account that flame thickening cold be applied just near the flame front, so a flame sensor is built and activates the thickening factor.
Flame sensor proposed is based on a "Arrhenius-like" expression:

$$\Omega = Y_F^{\nu_F}Y_O^{\nu_O}\exp\left(-\Gamma\frac{T_a}{T}\right) \tag{4.52}$$

where Γ is a factor which decreases the activation temperature. This parameter is very sensitive and dependent on the studied case. Therefor the present work is based on the formulation presented by Durand and Polifke [11], that considers the progress variable to define flame position:

$$\Omega = 16\left[c(1-c)^2\right] \tag{4.53}$$

where c is the progress variable, defined with instantaneous fuel mass fraction Y_F and the stoichiometric (or equilibrium) fuel mass fraction Y_F^{st}:

$$c = 1 - \frac{Y_F}{Y_F^{st}} \tag{4.54}$$

Sensor Ω controls the thickening factor through the function:

$$F = 1 + (F_{max} - 1)\Omega \tag{4.55}$$

G-equation

This approach is quite different from the previous one and was presented first by Williams [40].

The model consists on reducing flame front thickness to zero and describing its propagation with an equation based on field variable $\widetilde{G}$. Flame position is defined where $\widetilde{G} = G^*$ and the flame propagation's equation is defined as:

$$\frac{\partial \overline{\rho}\widetilde{G}}{\partial t} + \frac{\partial \overline{\rho}\widetilde{u}_i\widetilde{G}}{\partial x_i} = \rho_0 \overline{s}_T |\nabla \overline{G}| \tag{4.56}$$

The equation needs a closure for the subgrid scale flame speed s_T that must be modeled. As already presented for ATF model there are different methods to calculate this different velocity based on laminar flame velocity.

Flame surface density

Last presented approach is based on filtered balance equation on a scalar that can represent flame position and propagation [25]. This scalar can be progress variable c, mass fraction or temperature. Applying the filter on progress variable balance equation yields to:

$$\frac{\partial \overline{\rho}\widetilde{c}}{\partial t} + \frac{\partial (\overline{\rho}\widetilde{u}_j\widetilde{c})}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\overline{\rho}\left(\widetilde{u_i c} - \widetilde{u}_i\widetilde{c}\right)\right] + \frac{\partial}{\partial x_j}\overline{\left(\rho D \frac{\partial c}{\partial x_j}\right)} + \overline{\dot{\rho}^c_m} + \overline{\dot{\rho}^s_m} \tag{4.57}$$

where second and third terms on the right-hand side can also be seen as flame front displacement:

$$\frac{\partial}{\partial x_j}\overline{\left(\rho D \frac{\partial c}{\partial x_j}\right)} + \overline{\dot{\rho}^c_m} = \overline{\rho s_d |\nabla \Theta|} \tag{4.58}$$

although the flame front is too thin to be resolved within normal mesh size, a new larger filter for the progress variable is used, so the flame is very thin and precisely positioned. Resulting filtered equation is similar to the G-equation but, compared with an arbitrary G-field, the progress variable has the main advantage: c and related quantities are physically defined.

Boger et al. [5]proposed a model to estimate the flame front displacement, using wrinkling factor Ξ, as defined before, and laminar flame velocity s_L:

$$\overline{\rho s_d |\nabla \Theta|} \approx \rho_u s_L \Xi |\nabla \overline{\Theta}| \tag{4.59}$$

where ρ_u is the fresh gases density.

4.4 Conclusion

In this chapter the principal methods and models for simulations were presented. The most used method in the industrial environment Reynolds Average Navier-Stokes was omitted because less interesting for this work.

The work is based on Large eddy simulation and the basic Smagorinsky model was implemented in the used code as first, the engines simulation are run with this model. However to upgrade the future simulation also dynamic procedure of Smagorinsky model was implemented (Germano model) and tested on simple geometries like bluff body flame. This model has better results but the computational costs are 30% higher than Smagorinsky, so after model validation the engines computation were run with Smagorinsky.
In case of combustion ATF model was implemented in the code, that uses Arrhenius law with balance equation for every considered species. Although the engines are computed with air/fuel perfectly premixed, in the intake stroke after combustion, residual in-cylinder exhaust gas and fresh air are mixed together. Therefor dynamic version of ATF was implemented so this mixing process results unaltered.

Chapter 5

Numerical procedure

The mathematical models presented in the previous Chapter 4 need to be implemented in the working code. Since 3-dimensional flow solver cannot solve differential equation, those must be discretized and adapted into the code. This part is one of the most important and time consuming phase of the entire work.
This chapter is based on the description of methods needed to implement the presented model into the code and a briefly description original method used in this code.
Used code is KIVA-4MPI, software developed by Los Alamos Laboratories and obtained as source code by EKT (Energie und Kraftwerkstechnik) institute to simulate combustion inside IC engines with moving mesh. The code was developed originally for commercial use, therefor basic simulation method is RANS (Reynolds Average Navier-Stokes). Few years before this project begins first LES in the previous version of the code (KIVA-3V) was implemented by Goryntsev [15]. This work upgrade LES module from KIVA-3V into KIVA-4MPI and implements another LES model for flow, one for combustion simulation and other few modules needed for little improvement necessary to have a good quality the simulation.

5.1 Finite volume method

This approach is the most used in CFD codes. The entire computational domain is divided into finite volumes. In each of them, governing equations are solved. Therefore some modification inside equation must be done. Considering for example a scalar governing equation 3.11 integrated over cell volume ΔV:

$$\int_{\Delta V} \frac{\partial \rho Y_m}{\partial t} dV + \int_{\Delta V} \frac{\partial (\rho u_j Y_m)}{\partial x_j} dV = \int_{\Delta V} \frac{\partial}{\partial x_j} \left[\rho D_m \frac{W_m}{W} \frac{\partial X_m}{\partial x_j} \right] dV + \int_{\Delta V} \dot{\rho}_m^c dV + \int_{\Delta V} \dot{\rho}_m^s dV \tag{5.1}$$

To consider convection and diffusion terms as fluxes over cell surfaces, Gauss-Integration formula [39] is used:

$$\int_{\Delta V} \frac{\partial (\Phi)}{\partial x_j} dV = \int_{\Delta A} \Phi n_j dA \tag{5.2}$$

Furthermore time derivative will be written outside integral, considering almost stationary time step. In the next one every volume face is conventionally sorted with six different direction, northern, eastern, southern, western, fronter and rearer. For every face a normal flux F_f and a surface A_f is defined. Since every cell has one face in common with a

neighbor, surface and fluxes must be the same for both. Fluxes result conventionally inverted into two cells equations.
So long no approximation was used, but in the next step the most important will be done. Considering that every element is small enough to have no variation of properties inside its volume one could define mean $\rho^{\Delta V}$, $Y_m^{\Delta V}$, $\dot{\rho}_m^c$ and $\dot{\rho}_m^s$ over the cells. With these approximations one can write equation 5.1 into:

$$\frac{\partial}{\partial t}\left(\rho^{\Delta V} Y_m^{\Delta V}\right) + \sum_{n,e,s,\ldots} F_{f,C} = \sum_{n,e,s,\ldots} F_{f,D} + \dot{\rho}_m^c \delta V + \dot{\rho}_m^s \delta V \tag{5.3}$$

where:
$F_{f,C}$ denotes convective fluxes across the face f
$F_{f,D}$ denotes diffusive fluxes across the face f.
Important to say is that the last approximation can be used only if every cell inside the calculated domain satisfied the request of homogeneity for every considered property. Cell dimension depends on which kind of approach is used (RANS, LES, DNS), kind of physic solved (flow, chemical reactions, two phase flow) and cell positioning (middle of domain, near wall, near chemical reactions).

5.2 ALE scheme

Kiva is based on the Arbitrary Lagrangian Eulerian (ALE) method [18], [32]. As the name said this method is a mixture of Lagrangian and Eulerian standard methods and is used in KIVA-4MPI because it enables grid motion.
This approach needs definition of a grid subdivided in cell volumes, the corner of which are the vertices. It is convenient to define auxiliary cells centered about the vertices. Those are called momentum cells and are used to solve differential momentum equations. Velocity definition and storage will be done in cell vertices. The advantage is no interpolation is needed while nodes are moving. All the thermodynamic quantities are defined at the cell center, those if needed will be interpolated on the cell faces.
During the computation there are three phases, first two (A,B) are Lagrangian, last one (C) is Eulerian.

Lagrangian phases: A and B

In these phases will be calculated:

- Mass density equations for species and continuity
- Source terms for chemistry and spray
- Momentum equation (without convection terms)
- Internal energy equation (without convection terms)
- Other equations like k and ϵ if RANS activated

Eulerian phase: C

In phase C grid rezone will be done, the code interpolates velocities in the real vertice position, convective transport associated with moving mesh will be calculated. This is accomplished in a subcycled, explicit calculation using a timestep Δt_c. The timestep must satisfy the Courant condition:

$$\frac{u_r \Delta t_c}{\Delta x} < 1 \tag{5.4}$$

where u_r is the fluid velocity relative to the grid velocity.
Convection scheme can be two:

- quasi-second-order upwind (QSOU) differencing scheme
- partial donor cell (PDC)

both described in [2]

5.3 Time discretization

Timestep Δt is calculated at the beginning of each cycle and kept constant during its computation. Since the first phases are implicit no big restriction are required, but for accuracy reason and due to convective term calculation several restriction conditions must be applied.
First accuracy condition:

$$\frac{Du}{Dt} \Delta t^2 < f_a \Delta x \tag{5.5}$$

where f_a is a factor of order unity and Δx is an average cell dimension. This condition arises because terms of order higher than Δt are ignored in vertex position calculation.
Second accuracy condition is:

$$\lambda \Delta t < f_r \tag{5.6}$$

where f_r is of order unity and λ is an eigenvalue of the rate of strain tensor. This criterion limits the amount of cell distortion that can occur due to mesh motion. Other two criteria are introduced due to interaction between flow: chemical heat release and mass and energy exchange with the spray.
The convection timestep, based on the Courant condition is another stability condition that must be applied.
Other conditions are defined from the user, for example: the maximum timestep, maximum timestep based on crank angle engine, the maximal timestep increment between two cycles, timestep reduction due to incoming instability like valve opening or ignition.
All those condition are included on the timestep calculation at the beginning of every cycle and the timestep is the minimum of all previous presented.

5.4 Test filtering

During Germano model implementation, test filter equation 4.23 is applied and must be implemented. This filter is considering neighbor cells around the analyzed element and can be discretized with a weighted average like:

$$\widetilde{a}(x) = \frac{1}{nk_{tot}} \sum_{i=1,n} k_i a_i(x) \tag{5.7}$$

where the sum is done through all neighbor cells, k_i is a factor depending on the neighbor cell position, k_{tot} is the sum $\sum_{i=1,n} k_i$. In the standard case two or three level of neighbor cells can be chosen. First level of neighbor cells are the northern, eastern, southern, western, fronter and rearer cells those have a face in common with the analyzed element. Second level neighbor cells are element with an edge in common with the analyzed cell. Third level neighbor cells have just one vertex in common with analyzed element.
Analyzed elements have weight $k = 8$, first neighbor $k = 4$, second neighbor $k = 2$ and third neighbor $k = 1$. Implementing this function into ALE scheme based code need two different routines, one for the vertex based properties and a second one for cell center based quantities.
One issue must be solved for this implementation which is the boundary cell treatment. Those must be excluded from the filter and keep original values. Other problems can also be in case of parallel code, how to treat cells near CPU domain boundary where information are partially communicated. In this case one can consider the ghost cell layer on the boundary, this layer contain informations of the first domain layer of the neighbor CPU. On CPU boundary cells the average can be calculated without any problem, but in case of missing informations the average will be truncated just not considering those cells in the average.

5.5 ATF filtering

Another implementation issue appeared during wrinkling factor implementation is the estimation of subgrid scale velocity through the Laplacian of the resolved vorticity equation 4.51.
Implementation method was suggested by Durand and Polifke [11]. Since test filter is defined by a combination of spatial operators, those can be interchanged and the operator is:

$$OP(u) = c_2 \Delta^3 \nabla^2 (\nabla \times u) \tag{5.8}$$

where the vorticity can also be calculated as:

$$\nabla \times u = \begin{pmatrix} w_y - v_z \\ u_z - w_x \\ v_x - u_y \end{pmatrix} \tag{5.9}$$

Now applying Laplacian on equation 5.10 yields to:

$$OP(u) = c_2\Delta^3 \begin{pmatrix} (w_y - v_z)_{xx} + (w_y - v_z)_{yy} + (w_y - v_z)_{zz} \\ (u_z - w_x)_{xx} + (u_z - w_x)_{yy} + (u_z - w_x)_{zz} \\ (v_x - u_y)_{xx} + (v_x - u_y)_{yy} + (v_x - u_y)_{zz} \end{pmatrix} \tag{5.10}$$

A discretization of the second order derivative is applied. Since Colin [8] suggests to apply the operator with a step size double as the mesh size, every cell will be analyzed with neighbor cell on the same direction neighbor cells. Issue in this case is bigger than previous because on CPU boundary just one layer ghost cell is defined, so the information is missing for those cells.

Suggested solutions are: increasing the number of ghost cell's layer or reduce locally the step size from two to one. Second solution was applied in this work. Those kind of cells have normal computation for not boundary direction but reduced step size on the direction of the boundary layer, where ghost cells are included in the calculation and not its neighbor.

5.6 Ignition modeling

Ignition modeling in KIVA-4MPI is done through energy deposition as source term for the energy equation. During ignition the energy source term is calculated and added to the energy equation like the source term calculated during chemical reactions.

The model was already implemented in the standard code but the function to define in which cells energy is deposited was very slow and not precise, because ignition cell was supposed to be like cells for injection. Results impossible to define ignition position different from injection one, besides if injection is deactivated, ignition can't be defined.

To improve the code and neglect these problematics a new routine for position definition and ignition application was written, however basic concept of energy deposition remain the same.

First of all positioning routine is called during setup, here based on ignition coordinates, defined in the input file, is verified if cells are included inside this domain or not. Just cell centers in this case is analyzed. Included cells are saved in a vector and the total number of them saved as integer.

During computation chemical reaction routine is called every cycle and verify if timing for ignition is reached or not. When it is reached a loop for all the saved cells adds the source term to energy equation.

Chapter 6

Bluff body flame

This simulation is a test case already studied by Giacomazzi et al. [14].
Target of this simulation was to validate the models implemented in KIVA-4MPI for engines simulation following.
Implemented and tested models in this case are: Smagorinsky and Germano as SGS models, ATF as combustion model with dynamic thickness. As SGS model verification non-reacting cases will be presented, after that combustion will be activated in combination with Smagorinsky model.
Although the geometry is simple, interaction between flow and bluff body create a complex recirculation volumes where complex turbulence phenomena are created.

6.1 Configuration

The configuration consists of a straight channel with rectangular cross sections (Figure 6.1). The triangular bluff is body located $5.5cm$ after inlet has flame anchor function in reactive case, or a simple obstacle and turbulence source for non-reactive case.

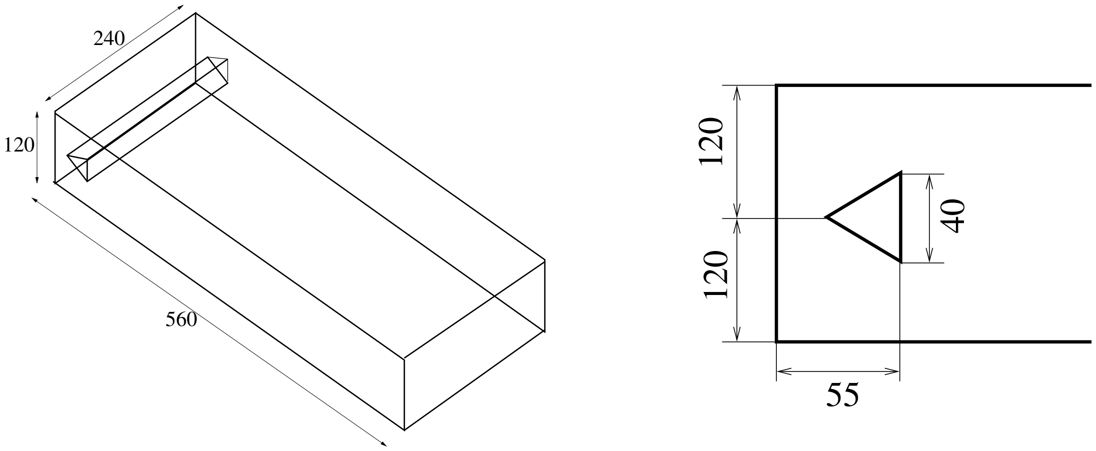
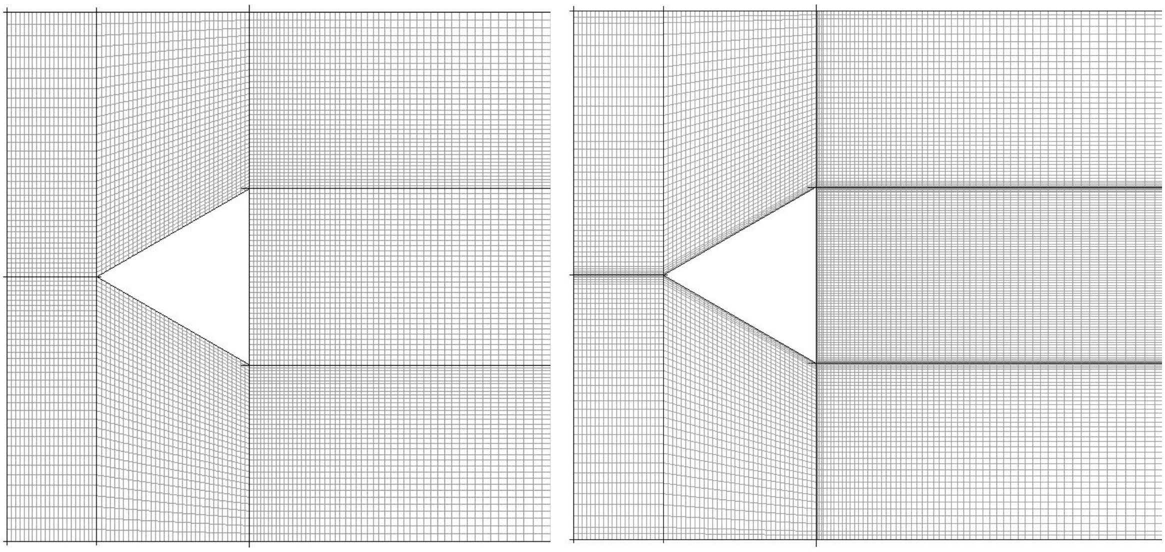


Figure 6.1: Geometry configuration

Fresh mixture of air and propane with mixture fraction $\phi = 0.65$ is inlet boundary with $17ms^{-1}$ and $288K$.
Fixed pressure non-reflecting boundary condition is used as outflow and all the walls have

no slip properties. Non-reacting simulations were run with subgrid scale model Smagorinsky and Smagorinsky with Dynamic procedure (Germano) for at least $1.0e^5$ time steps with $1.0e^{-5}$ second every time step for two different grid of 900k and 1.4M cells, those are shown in Figure 6.2.

As shown in Figure 6.2b the refining was applied behind the bluff body, in the recirculation zone and near walls. Both grid were calculated on 12 cores that perform 6 seconds per time step on first grid and 8 seconds on refined grid appling Smagorinsky model. Germano model needs 30% more time. The configuration was also reproduced experimentally by Sjunnesson et al. [36] where the results of average velocity, average fluctuation and temperature where extrapolated. Results are reported in four different sections after the end of the bluff body that became zero coordinate point for stream wise direction (Figure 6.3). On the selected sections data are collected and compared with experimental data. Statistics collection is done during computation with an implemented



(a) Slide of coarse grid **(b)** Slide of fine grid

Figure 6.2: Computed grid

Figure 6.3: Measurement section positions

routine, because of the limited disk space was chosen to not write all data but just average velocities and fluctuations to every selected point. Mean velocity is calculated as follow:

$$v_{avg}^{n} = \frac{v_{avg}^{n-1}(n-1) + v^{n}}{n} \tag{6.1}$$

where v_t is the instantaneous value of the velocity or velocity component, n is the number of collected data, in this case the number of time step included on statistics, v_{avg}^{n-1} is the average value calculated until previous time step. With this kind of equation just one value is saved during computation.
Same proceeding was used to calculate rms values, defined as:

$$v_{rms} = \sqrt{\frac{(v_{temp}^{n-1})^2(n-1) + (v^n)^2}{n} - v_{avg}^{n}} \tag{6.2}$$

where v_{temp}^{n-1} is the average square value of the velocity, defined as below:

$$v_{temp}^{n} = \frac{(v_{temp}^{n-1})^2(n-1) + (v^n)^2}{n} \tag{6.3}$$

6.2 Validation

Code validation is done with only the main velocity component and its fluctuation because this data were available on the experimental results.
The following plots compare Smagorinsky and Germano model applied on coarse grid with experimental results.
In Figure 6.4 on first plot one can see how this two models have some differences, speaking about velocity Germano model fits better the curve in the first section near the wall, where Smagorinsky model has a bigger negative velocity as recirculation. Furthermore in the second section the recirculation is not really good captured from both models and velocities in the middle is too high compared with experiment, if no better results are obtained will be problems in the fired case where the fresh air is not proper entering on the recirculation zone. Third plot shows how Smagorinsky model results more dissipative as Germano and experiment, here for the first model recirculation zone has quite small effect but experiments show how the velocity on the middle are still smaller as on sides, also in this case Germano model is giving better results. On the last section both models are really close to each other and velocity is circa 5% over experiment.
Another quite important difference that can be seen overall is that Smagorinsky model near the wall has a bigger velocity gradients as Germano, this happen because the viscosity with first model is calculated with a constant value overall on the other hand with dynamic procedure is decreasing near the wall reducing viscosity in this zone.
Figure 6.5 shows stream wise component of velocity fluctuations for the same case. What was noticed before is also here repeated, Germano model is closer to the experimental results and have a better caption of fluctuations on recirculation zone, especially in the first section near the bluff body. Some lack of precision there is also for Germano model on

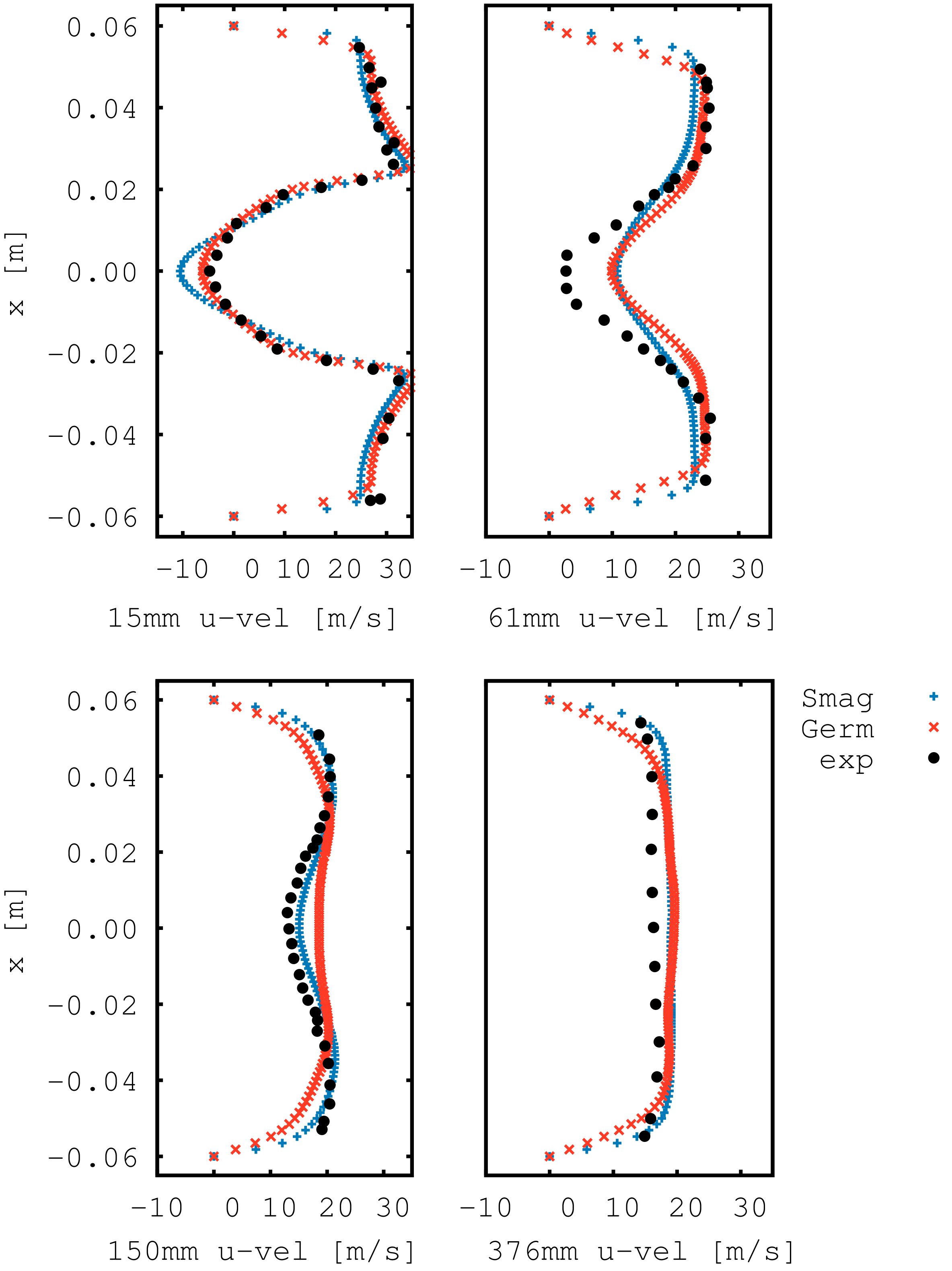


Figure 6.4: Axial component of mean velocity plot on the four different positions. + simulation with Smagorinsky model, **x** simulation with Germano model, ∘ experimental data

second and third section where average fluctuations are too high, this can be induced by the coarse grid. Nevertheless also statistics collection can also play a role in this case. The statistics must be collected on enough wide interval of time step and physical time. On the other side initial solution cannot influence the final result, so before starting with statistics collection the simulation needs to run enough physical time to reach developed flow. On both simulations statistics collection starts after 30.000 time steps (ca. $0.15s$) when the flow starts with $17ms^{-1}$ has covered on average $2.55m$ that should be enough to develop the flow. Statistics collection was computed for at least 50.000 time step (ca 0.25 sec) also in this case enough for statistic. Second analysis is done to verify grid independence. It is already known that LES has different solution for every different grid because solved eddies are more if filter dimension becomes smaller, so converged solution definition doesn't exist like in RANS context. However as explained from Pope [30] at least 80% of turbulent kinetic energy must be solved to have a proper large eddy simulation, valid for all the parts of simulation comprehensive near wall. To verify that this requirement is satisfied the follow parameter, proposed by Pope [31], can be calculated:

$$M(x,t) = \frac{k_r(x,t)}{K(x,t) + k_r(x,t)} \tag{6.4}$$

where, K(x,t) is the resolved turbulent kinetic energy:

$$K(x,t) = \frac{1}{2}\langle (W - \langle W \rangle) \cdot (W - \langle W \rangle) \rangle \tag{6.5}$$

and $k_r(x,t)$ is the turbulent kinetic energy of the residual motion, calculated in according with Chai and Mahesh:

$$k_r(x,t) = 2C_l \Delta^2 |S|^2 \tag{6.6}$$

$$|S| = \sqrt{2S_{ij}S_{ij}} \tag{6.7}$$

Parameter M value are between 0 and 1: $M = 0$ corresponds to DNS and $M = 1$ to RANS. To reach at least 80% of simulated structures:

$$M(x,t) \leq 0.2 \tag{6.8}$$

if M is grater than this value the grid must be refined and if is smaller is better to coarse because too much structures as needed are solved.

This parameter verification was done also for this case, but just on analyzed sections and on a limited number of samples because data collection was done in post processing and not during the calculation. Instead of many thousands of sample just hundreds are collected. The verification is done to understand if grid dimensions are sufficient small or the grid needs refinement, for that case too precise statistic is not needed.

Figure 6.6 shows plots with calculated values M in previously selected sections, computed on the coarser grid and Smagorinsky model. Although the information is limited because is extracted just on the four sections, one can see that two areas have M grater than 0.2: the recirculation zone in the center especially near bluff body surface and near external walls area where values are always too high. For those reason it was decided to refine the grid behind the bluff body and near the wall as shown in 6.2b. Comparing results of two different computed grids with Smagorinsky model and a comparable number of statistic

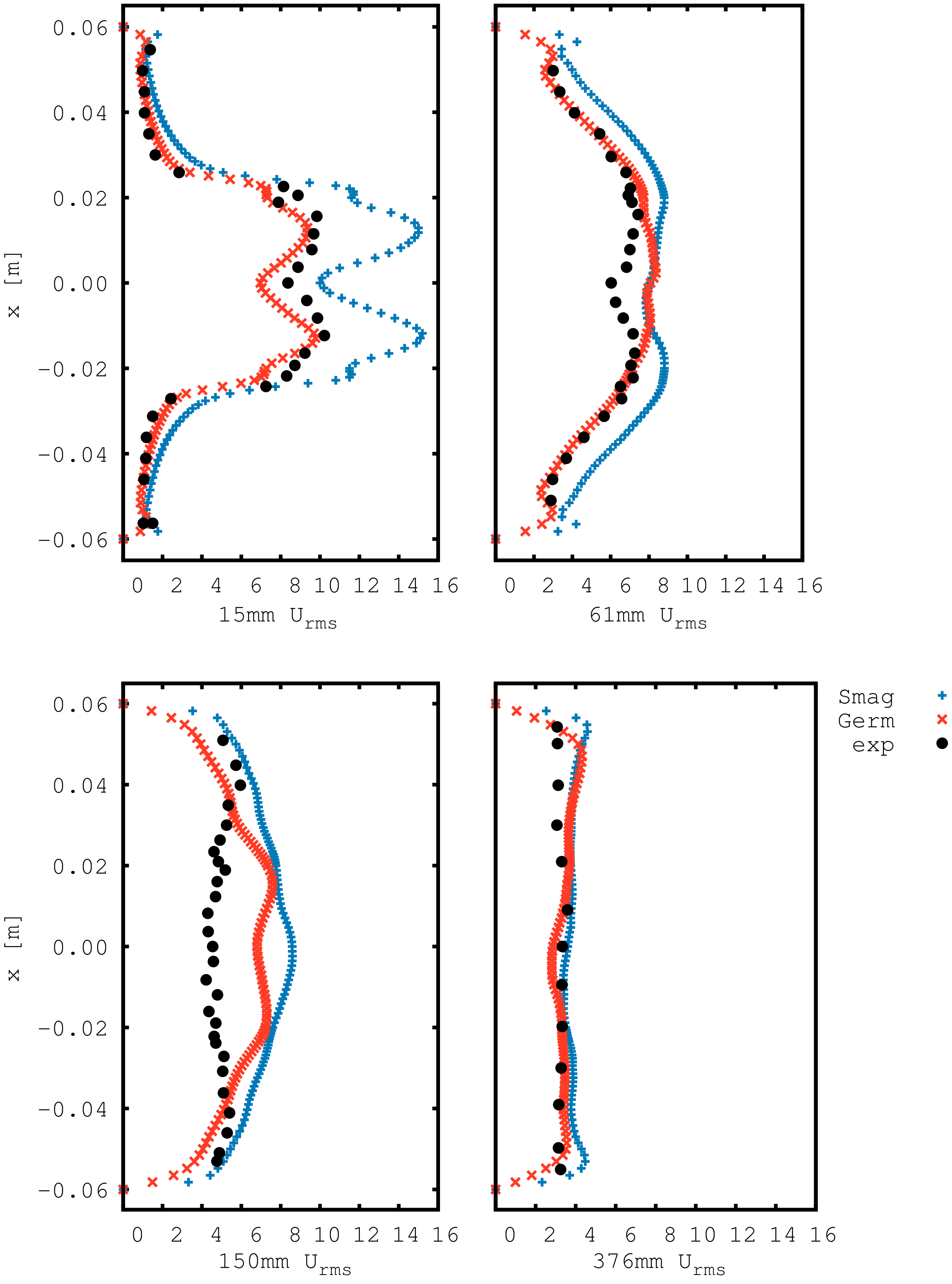


Figure 6.5: axial component of root mean square of velocity plot on the four different positions. + simulation with Smagorinsky model, **x** simulation with Germano model, ○ experimental data

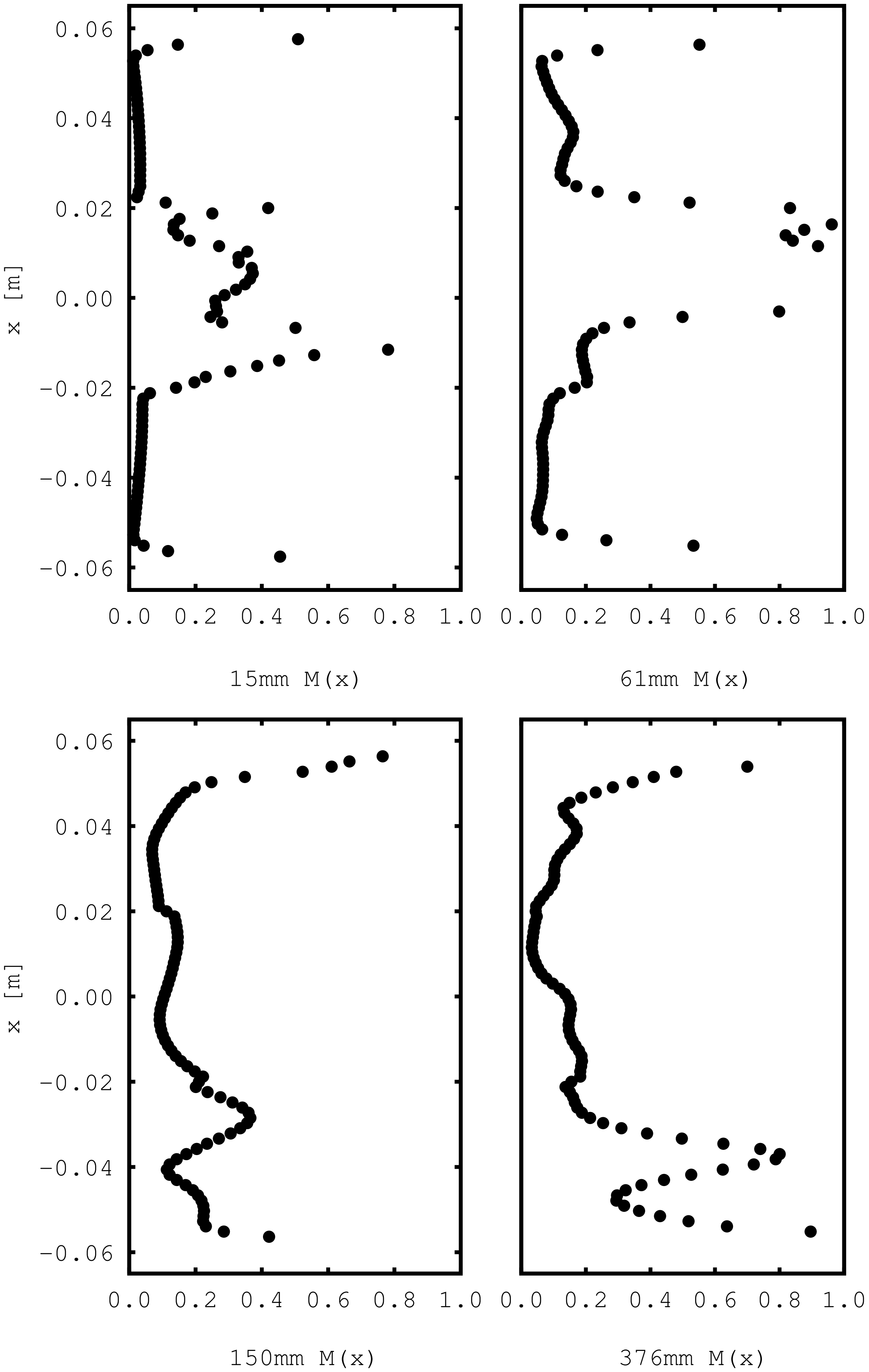


Figure 6.6: Parameter M calculated on the same position of collected results

convergence. Direct comparison is shown in Figure 6.7 where mean velocity are plotted into selected sections. Mean velocities are slightly better than previous grid in particular on second and third sections where coarse grid didn't predicted well. On the other side, the fluctuation plot in Figure 6.8 shows that the physical computed time wasn't enough to have a proper statistic on the fluctuations.

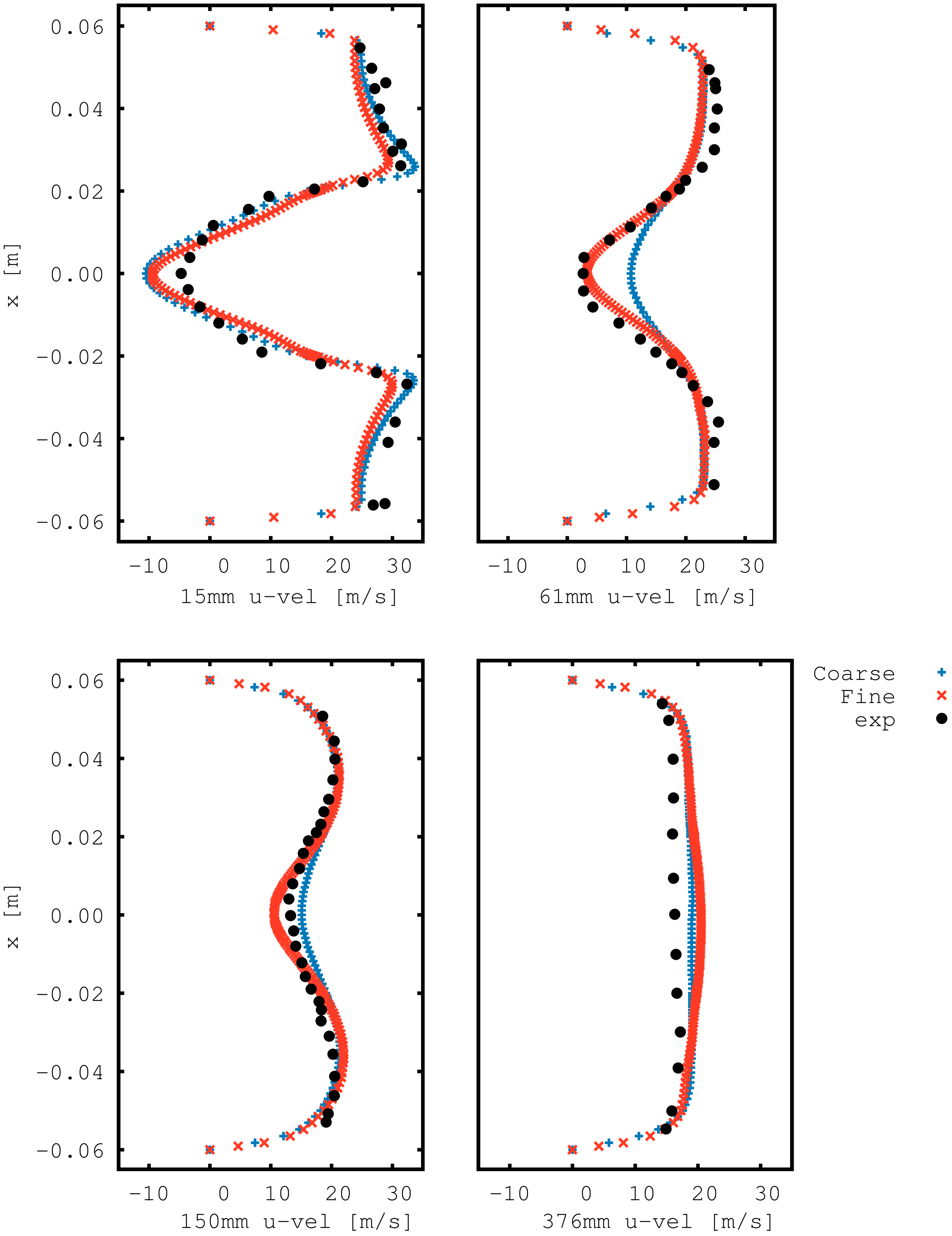


Figure 6.7: comparison between coarse and fine grids mean velocity

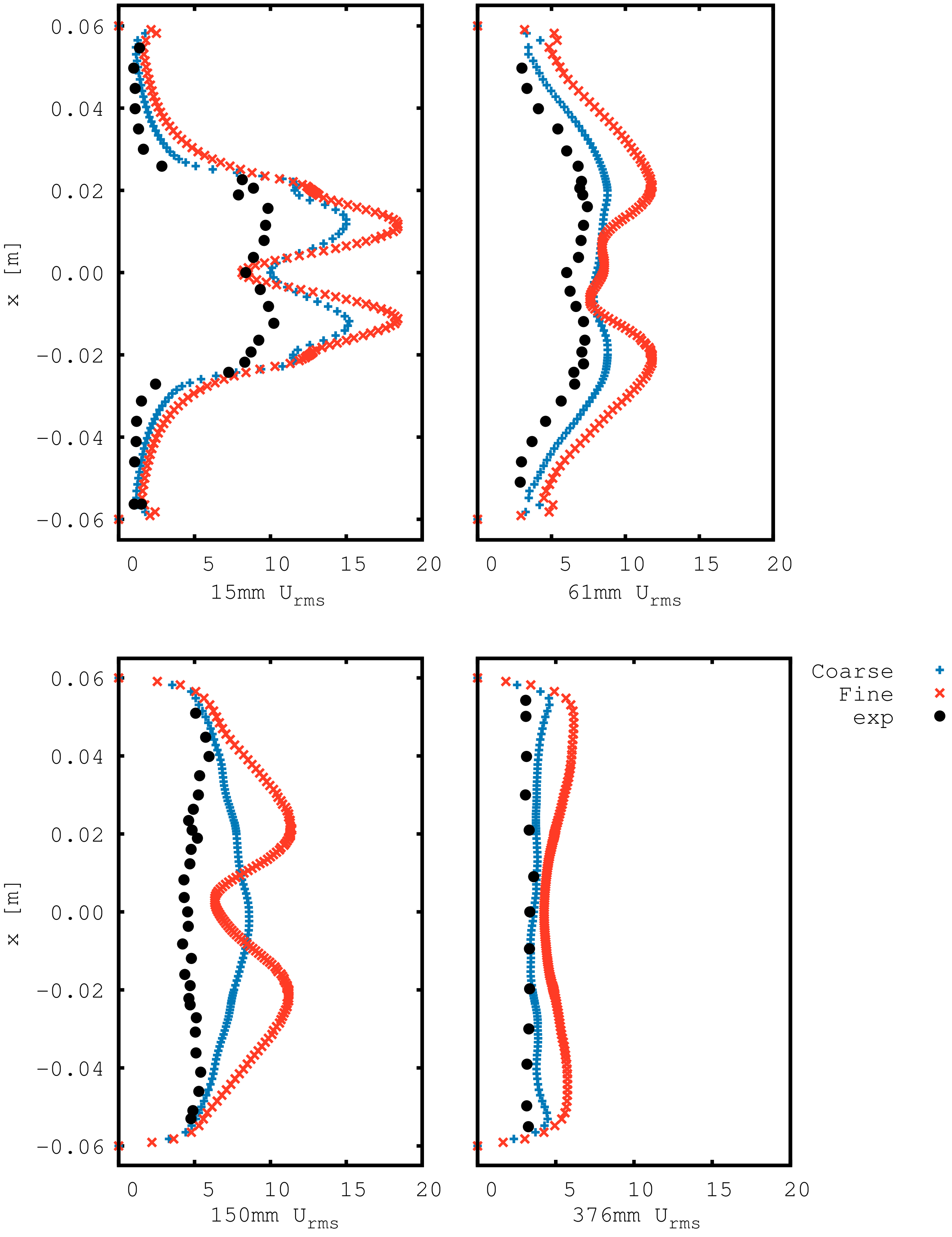


Figure 6.8: comparison between coarse and fine grids: mean fluctuations

6.3 Fired Case

In this last section, fired case simulation problematics will be presented. The configuration is the same as discussed previously, only difference is the combustion reactions activation and the mixture ignition at the beginning.
Activated chemical reactions are following:

$$\begin{aligned}
C_3H_8 + 5\,O_2 &\longrightarrow 3\,CO_2 + 4\,H_2O \\
O_2 + 2\,N_2 &\longrightarrow 2\,N + 2\,NO \\
2\,O_2 + N_2 &\longrightarrow 2\,O + 2\,NO \\
N_2 + 2\,OH &\longrightarrow 2\,H + 2\,NO \\
N_2 &\longleftrightarrow 2\,N \\
O_2 &\longleftrightarrow 2\,O \\
H_2 &\longleftrightarrow 2\,H \\
2\,CO_2 &\longleftrightarrow 2\,CO + O_2 \\
2\,OH &\longrightarrow O_2 + H_2 \\
2\,H_2O &\longrightarrow 2\,H_2 + O_2
\end{aligned}$$

Fuel combustion will be simulated with one step chemistry, represented by the first reaction, the remaining nine reactions are included to simulate intermediate species and reduce the amount of heat produced by first reaction alone. Temperatures and flame front velocity were verified before 3-D simulation through 1-D flame simulation as routine for every new tested chemistry.
Ignition is activated after 3000 time step so the cold flow can first develop and the recirculation zone will be established. Ignition at the simulation begin can be dangerous for flame propagation and destabilize the computation.
After ignition the flame propagates until the end of the domain, this takes around 10.000 time steps before this time isn't possible to start statics collection because the flame results unstable and the flow velocities are developing. Statistics are collected from 20.000 time step from the begin of the simulation.
During various simulation statistics collection simulation became unstable and wasn't possible to complete this phase. The issue appears after a while the flame reaches outlet boundary condition, imposed as continuity outflow. This kind of boundary condition makes a balance of all physical components and impose a zero gradient through the boundary so the outlet results as a normal interface between fluid cells.
However this process seems to have some issue because when chemical species through outlet aren't the same during the simulation local pressure has a gradient, as consequence velocity field is changing locally. This issue can be caused by a wrong species balance, every chemical species change local mixture in every time step and every cell changing also cell density. Balance on outlet is done with cell density and if this results wrong to reach balance pressure or temperature must be changed in the domain cell. Therefor velocity near outlet are changed from continuity equation's routine.

Simulation runs stable but results on the last part of the domain present much higher velocity and fluctuation as experiments. This issue can't be solved but results far from outlet are good.
Figure 6.9 shows velocity plots in all four measured sections, results fit quite good near the bluff body where both model have a good match, also in $61mm$ section results fit quite good, but Germano model in this case has a reduced recirculation zone maybe cause by the bigger influence by outlet than Smagorinsky. Last two sections results aren't very good because of the strong influence to the outlet. Figure 6.10 presents just the results of the first two sections, where outlet's issue isn't influencing the flow and can be easy noticed that profiles are matching quite good with experiment.

6.4 Conclusion

This simulation was set up to validate code and new models for LES cold and fired, after results discussion can be resumed that this goal was partially reached. Non-reacting simulation have quite good match for both implemented models, although Germano model results much precise as Smagorinsky must be taken into account the increased requested computational time, at least 30% more that simple model, that reduce actual preferences for it.
Really important is to calculate different grids during computation because can happen that precision of the model is strong reduced by a too coarse grid, if implemented in code suggested criteria results very efficient and precise to verify filter dimensions.
On the other side fired case presents a lot of difficulties due to the boundary conditions, otherwise partial results obtained are a good point to start for more code development.

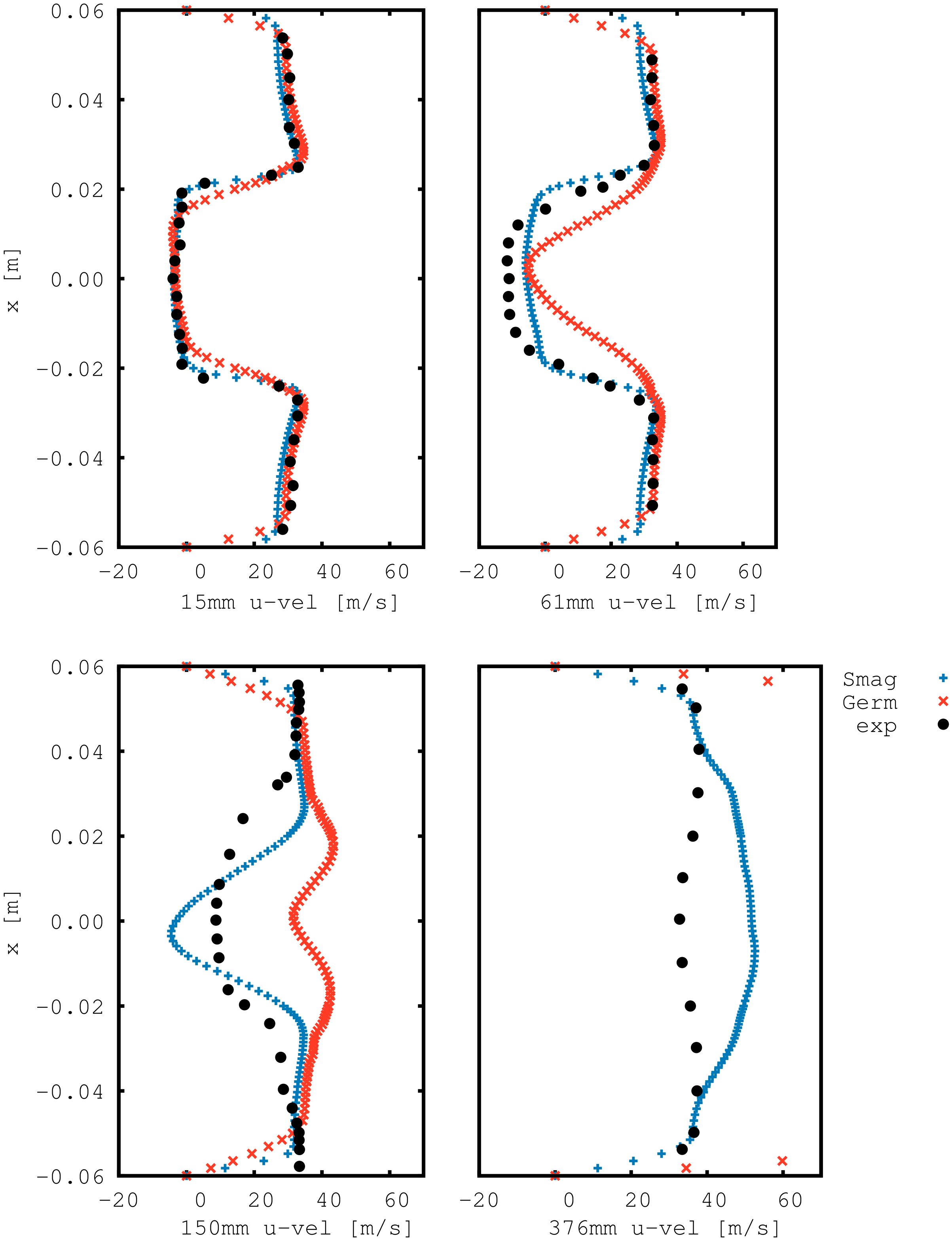


Figure 6.9: axial component of mean velocity plot on the four different positions. + simulation with Smagorinsky model, **x** simulation with Germano model, ○ experimental data

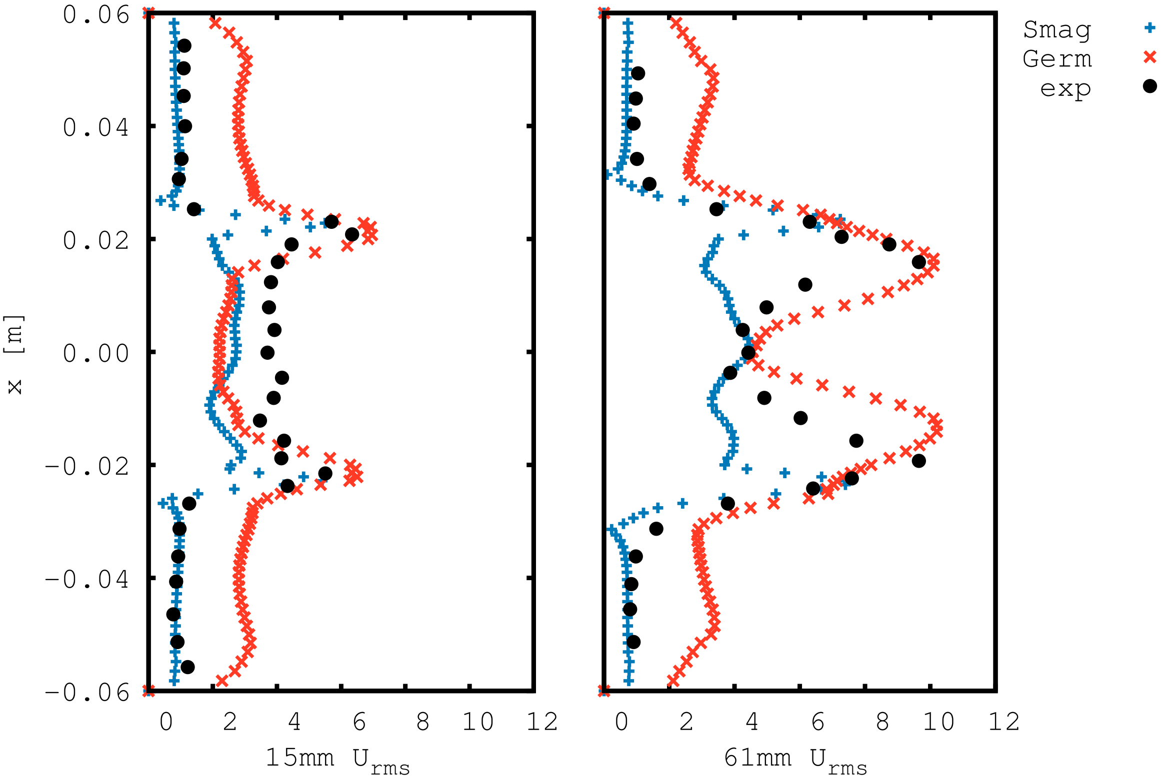


Figure 6.10: axial component of root mean square of velocity plot on the four different positions. + simulation with Smagorinsky model, **x** simulation with Germano model, ◦ experimental data

Chapter 7

Numerical simulation of engines

Grid generation for internal combustion engines CFD is one of the most important phase and sometimes the most expensive in terms of human resources. Speaking about unsteady simulation this process is really complex due to moving parts inside the domain like piston and valves, those move all together sometimes really close between each other. Lots of commercial software have an integrated pre-processing module to create easily a good quality grid without too much users difficulties. KIVA-4MPI hasn't a good quality integrated module to generate grids appropriate for LES analysis, so it has been decided to couple it with ICEM CFD, commercial software able to create an output file compatible with KIVA-4MPI. New KIVA version (KIVA-4MPI), the update of KIVA-3V, allows the use of o-grid and unstructured grid, after some experiment it was decided to use hexahedral grid because is much simple for the solver to manage a prism moving grid than other type of cell shape.

7.1 Mesh generation

Creating a functional grid with o-grid using ICEM CFD needs a different approach than the standard used in the past explained in Kiva-3V manual [1] or used by Goryntsev [15]. To have a better view of issues both methods will be explained in this section.

7.1.1 KIVA preprocessor K3PREP

Before two different ICEM CFD methods must be explained that KIVA has a pre-processing included module, but this is seldom used because of many limits given by it. K3PREP is a module included in KIVA package to mesh simple geometries. It isn't supported by a graphic interface that leads to big difficulties for the user, but in KIVA package are included some example that could be used as start point to create a similar discretized domain. Basically this module works with block structured grid, that is the easiest and efficient method to work with moving mesh in KIVA, needs an input text file that includes the list of all blocks and nodes needed to create the final grid. If the mesh is simple and small in terms of cell number, this file is relative short and easy to set, but in case of large grid or complex geometries that file became really difficult to compile because of the big number of input needed.

7.1.2 ICEM CFD block structure

ICEM CFD is a powerful mesh generation of ANSYS package and has the possibility to create structured grid manually and unstructured grid automatically. KIVA-3V was compatible only with blocking structured grid, so this method was the only possible solution. Grid generation procedure begins creating a single block including the whole domain, after that this will be split in so many blocks as the geometry needs to be matched. In Figure 7.1 this procedure is shown, also for simple geometries this procedure could be really complex and difficult to plan because of the big amount of needed blocks that the volume needs to be decomposed.

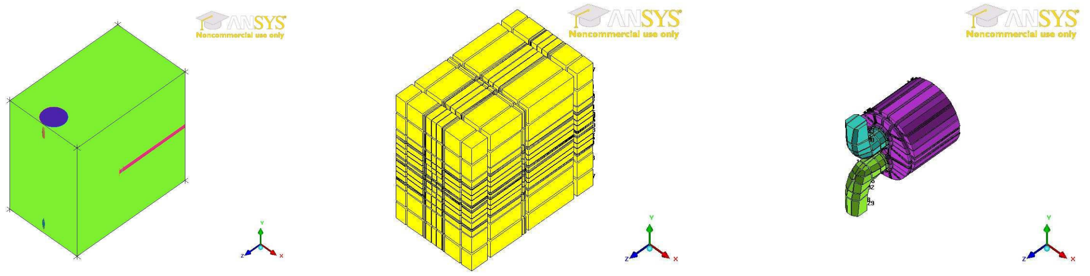


Figure 7.1: Blocking structure grid creation's initial steps

Second step is an iterative procedure (Figure 7.2) to find the right balance between quality and dimension of cells, basically more cells there are much more are angles dissimilar to 90 degree, that lead to a cell quality issue. Typical example of that are the cell on the top of the valve stem, because the blocking structure impose two different blocks, but because of the pipe curve two edge created to be normal to each other became quasi parallel and the resultant cell is distorted. So that happen on liner were four block corners must be adapted to the round shape of this component. On those distorted cell the solver has a lot of problem and often if the calculation stop the post processing shows that an instability started from a low quality cell and propagate time step after time step on the entire domain.

Figure 7.2: Blocking structure grid creation's iterative procedure

This kind of disadvantages makes the method not flexible and sometimes also unstable, becoming a big obstacle for LES analysis where lot of adjustments and refinements to increase cell's quality are needed. One example suggesting people on the institute EKT TU-Darmstadt to develop a new mesh generation method, was an analysis of a really simple engine (TCC-GM configuration) with flat piston and two parallel valves. Because of the particular shape of pipes it was difficult to create a mesh bigger than 100.000 cells and that was not enough to make a properly LES simulation, the results shown a totally absence of cycle-to-cycle variation typical for LES simulated phenomena.

7.1.3 ICEM CFD mixed method

As already mentioned this method was develop in EKT TU-Darmstadt institute together with K.Nishad , T.Breitenberger to face problematics included in the old proceeding. Grid generation with this process isn't easy or faster than before, but grid quality and flexibility are increased as results. This method also includes the possibility to use o-grid, which is a really big improvement for grid's quality.

7.1.3.1 Piston meshing

The method begins with mesh generation of a single surface. Generally is the lowest point of the domain (piston on the bottom death center). This could be done using planar blocking and could include as many split as needed so as many o-grid as needed too. During this phase it is really important to take into account of the complete cylinder shape, including valves moving and head shape's projection. That means the valves shape must be drawn on the piston because the target is to have parallel cells layer in squish volume, so during piston motion there will be no other cell deformation than in piston moving direction. In case of parallel valves' configuration drawn profile on the piston is just a vertical projection of curves on the below surface, but if the geometry is canted other factors must be considered, like valve lift, angles and head shape. Usually in those cases one can just draw valve moving volume and its bottom will be projected on the piston, so from piston to valve bottom node's position has the same coordinate and there isn't going to be lateral cell deformation during piston motion. In case of really complex head geometry it must taken into account of that during piston grid generation (Figure 7.3), to exclude the possibility of other deformation but in motion direction during all the piston stroke.

All previous actions were done in planar blocking structure mode, basis of grid generation mixed method with ICEM CFD code, when this process is finished the grid will translated from blocking to real mesh, creating a surface mesh from blocking. The advantage of this mode is that is making every cell and nodes free so one could work deforming every cell, on the other hand blocking mode one could just move blocks edges and nodes, without controlling every single cell but having less trouble with geometry deformation. To resume: blocking mode is useful because just blocks have possibility to be deformed, but in case of complex geometry it could be really difficult to manage refinement in different part of the geometry. Mesh mode must be use with caution because of the increased number of degree of freedom but it has lots more functions to modify the grid, just for KIVA-4MPI is possible to create a moving mesh with o-grid that blocking mode doesn't have because

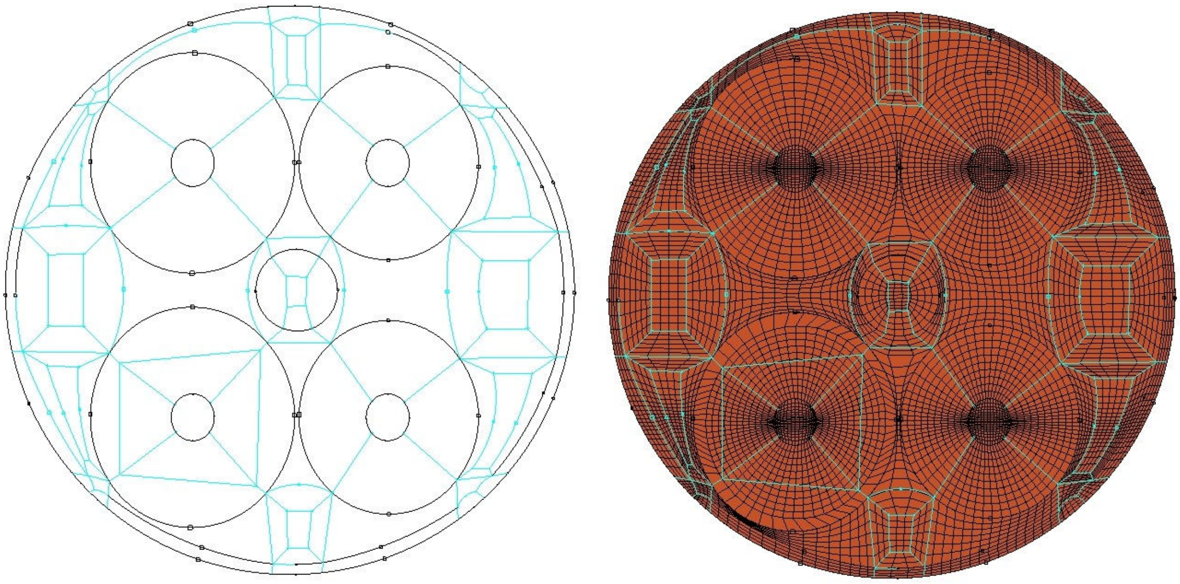


Figure 7.3: Piston mesh generation using blocking structure

the boundary condition are wrong exported from ICEM CFD to the input file for KIVA-4MPI. After transforming blocks in real grid a smooth could be done (Figure 7.4), that improves cells' quality keeping initial surface shape fixed, it's also important to take into account of all surfaces above, like valve edge projection or particular bowl on head shape, and freeze those nodes in their original position.

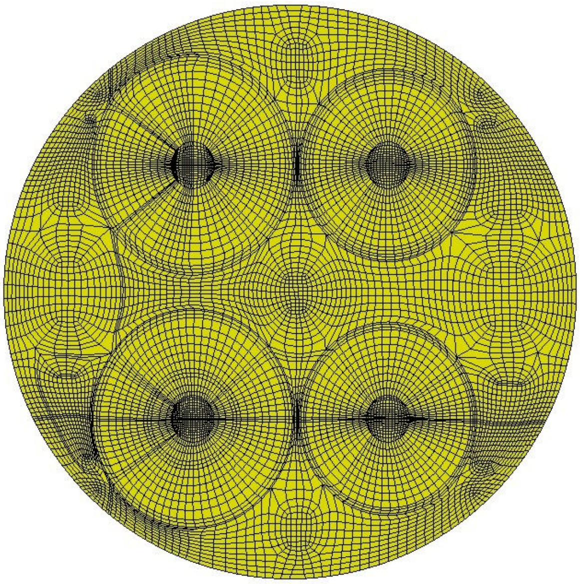


Figure 7.4: Piston mesh generation after smoothing

7.1.3.2 Bowl meshing

Optional mesh volume that could be useful is the bowl. Usually bowl is the concave piston part created to reduce TDC volume, helping spray and combustion processes to have a better efficiency. This piston shape is really common in modern engines development. In all the presented cases pistons are flat and this volume never appears. Although this volume is typically below the piston surface, in KIVA-4MPI it could be created above the surface and introducing a volume where the grid is fixed on piston's reference frame and have a better quality grid especially near TDC while combustion process is active.

7.1.3.3 Squish meshing

This is the simplest part of the procedure, starting from piston mesh one needs just to extrude the cells' surface until piston top position (Figure 7.5). After extrusion the grid could be arbitrary refined, usually above parts have much more cells that bottom one because the biggest flow velocities are concentrated close to valves. So long as valves have little lift there aren't problem, because piston and valves moving volumes never cross to each other, but if this is not true that could be a problem because of the particular KIVA-4MPI moving cells strategy, this requires a different cell rezoning that must be done in the code. About this particular issue will be spoken below.

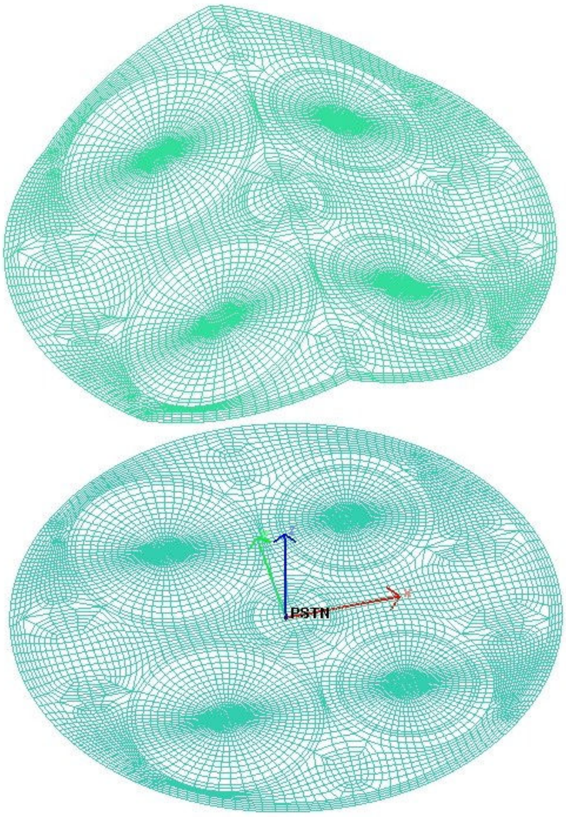


Figure 7.5: Squish generation through extrusion

7.1.3.4 Valve moving trace meshing

This is probably the most problematic and time consuming process during mesh generation, not just during discretization process, but also during moving process in the simulation. Sometimes a good quality cell is badly positioned during valve surface moving, is distorted and creates instabilities on pressure or temperature solution. This issue occurs because on contrary to piston that moves through the entire volume from wall to wall, valves are moving through a portion of that, and penetrate into the computational domain opening accesses for inflow or outflow. During those surface movements, new cells are activated and others are deactivated to leave space for valves volume. New cells must find an optimal position because around valve edges there are fluid cells that never be moved. Of course those difficulties are proportional to valves lift and number of cells activated during the simulation, so in case of LES more cells than RANS are requested and that kind of problem happen more often. Every kind of simulated engine has different difficulties, some engines have just 2-3mm valve lift some other more than 8-9mm, although for an optimal flow resolution it will be better to have circa the same amount of cells on valve maximum lift position, sometimes it's really difficult draw a grid with too small cells. As for piston moving also for valve moving a good grid creation in ICEM isn't enough for an optimal simulation, so sometimes is also necessary to change the original moving algorithm in the code. Explaining grid generation procedure: is really important that all cells below valves surface match the valve surface to neglect deformation on direction different to the moving one, Figure 7.6 shows a particular of the mesh where every valve has cells parallel to its moving direction. Every layer is normal to valve moving displacement and the cells matching with valve edges. For this reason valves surfaces were projected on the piston during first part of mesh generation. In case of canted valves, create this portion of volume require more steps, first of all valve shape extrusion along their moving direction, after that the rest could be extruded (Figure 7.7). This last step is really complex because of canted valve volumes that cannot move due to the previously explained issue, so extrusion direction is different for every portion of the

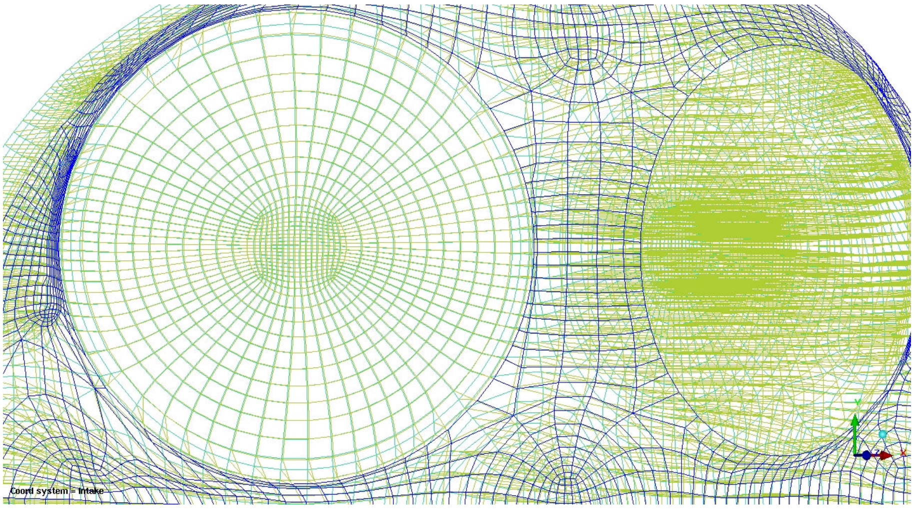


Figure 7.6: Particular of valve moving volume

geometry and that takes a lot of time because there isn't an automatic procedure to do it. Really important is also to keep a look on grid quality, because it's very easy to have inverted cell or just bad quality in this volume (see Figure 7.12). Cell quality in this region could make difference on the computational time. During intake and exhaust strokes the highest velocity are inside those cells, higher is velocity more unstable is the cell. If a cells has a good quality the issue aren't too big but in case of distorted cell the zone becomes unstable. So long the grid is composed of 2 vertical cells, the first from piston to lowest valve position interface, the second from that one to the head. Before proceed with the split that divide equally every edge, the valves must be shaped (Figure 7.8). First of all highest layer will be projected on valve surfaces, taking care that every edge stay parallel to moving direction. This layer will be head layer so it needs to be moved up

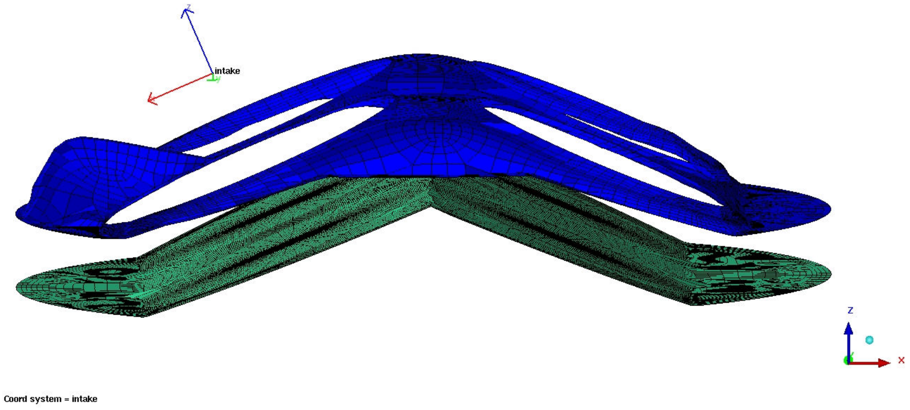


Figure 7.7: Head surface grid creation

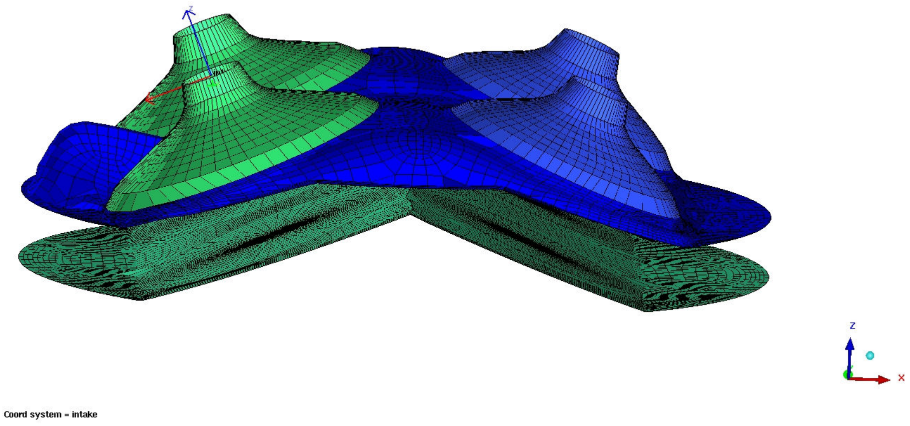


Figure 7.8: Valve top surface grid creation

of $0.1 - 0.3mm$. Now one proceeds with the first split, that must be really close to the highest layer. Between head and valve must be created one layer cells, this will be present also when valve are in closing position but those cells are deactivated. Then when valve is opening cell layer between head and valve is activated. This layer is very thin because in the real geometry this space doesn't exist so the software approximates valve opening (and closing) phase, of course lower this transition better the approximation.
Succeeding top valve also bottom valve will be created, layer thickness is decided from valve dimensions, typical values are between 2 and 3 millimeter. Since the split divides equally the edges, initially bottom valves shape will be similar to top surface, so every nodes must be projected to the geometrical to obtain the exact volume shape (Figure 7.9).

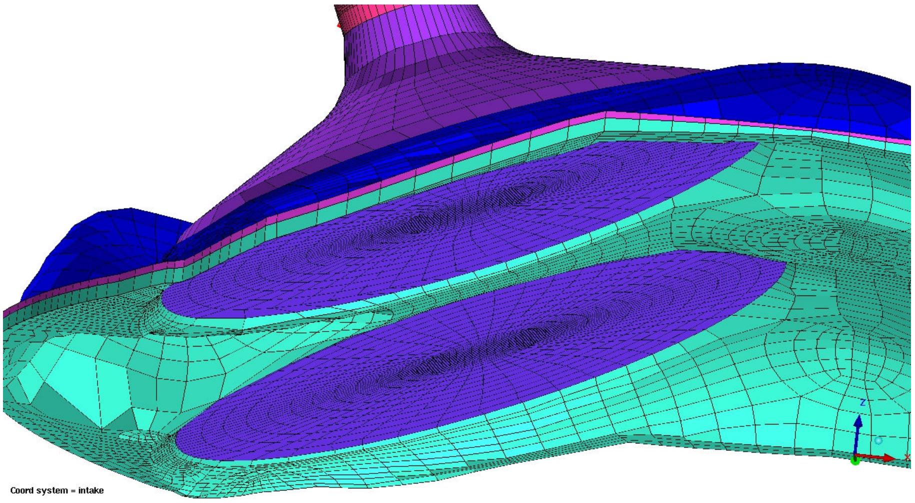


Figure 7.9: Valve bottom surface grid creation

7.1.3.5 Port meshing

There are three options for port meshing: continue with extrusion method and shape all this section by hand; return to standard method meshing ports separately with blocking grid; let the software creates automatic mesh generation. Last option require a software compatible with tetrahedral cells mixed with hexahedral ones, although KIVA-4MPI could handle this kind of grid as long the cells aren't moving isn't suggested to use this method.Other two options are both valid.
First method starts from the actual grid and extrude that. Sometimes there are problems drawing curves and particular points of the geometry, like cross between stem and port.
Second method has the possibility to insert o-grid inside blocking and quasi automatic proceed to create the grid. Domain is created in a new file, afterward needs to be imported and coupled manually with the combustion chamber.
Blocking method is the one used in this work presented cases. Proceeding is very simple because is just the block structured already used in [1], [15], the only difference is that one must take care of node's number and position on the interface to merge easily with combustion chamber. (Figure 7.10)

When both ports are ready they will be imported and extruding last combustion chamber layer and all regions will be merged together (Figure 7.11). This step must be done manually by merging node to node.
As last step of this procedure, is split vertical edges as much as the discretization needs.

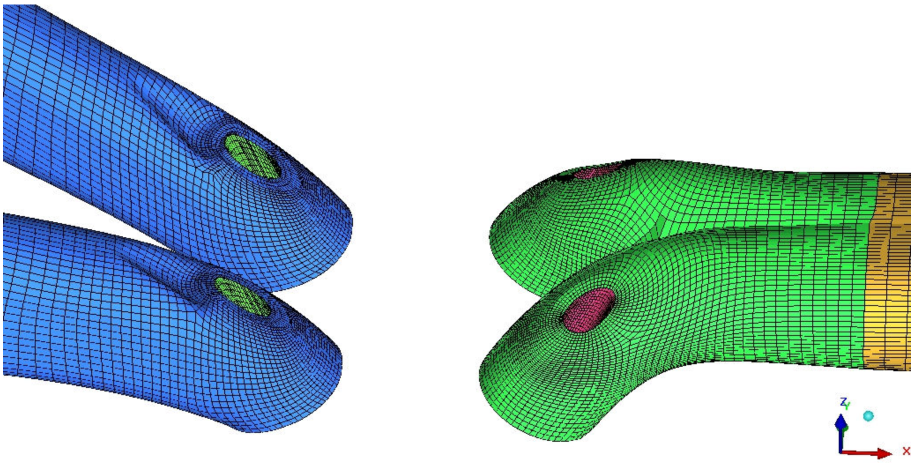


Figure 7.10: Ports grid creation in a separate file

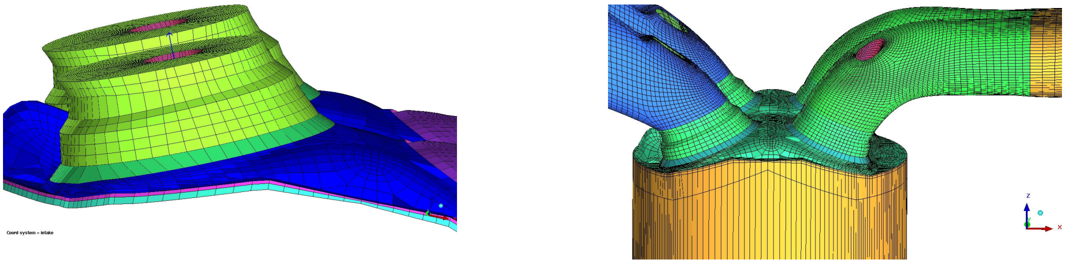
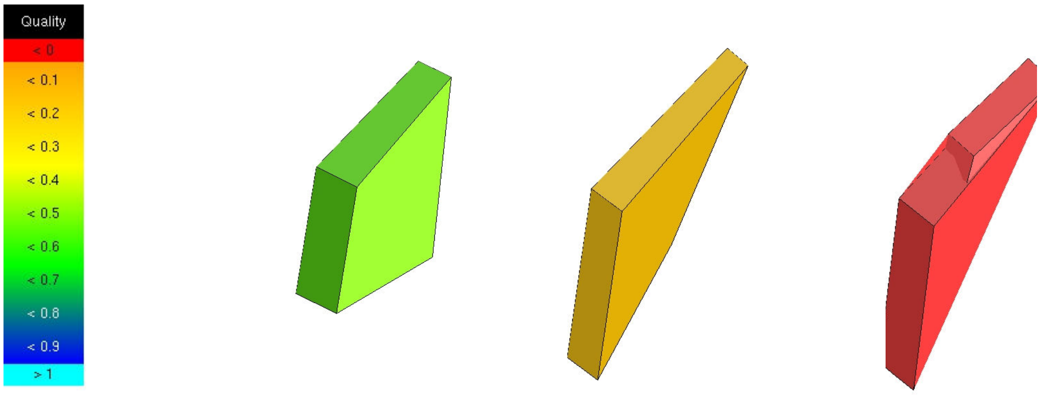


Figure 7.11: Ports and cylinder grid coupling

7.2 Mesh motion

As already mentioned above, after grid generation one can't be sure that simulation will work, sometimes if the grid has good quality and no inverted cell appears, mesh motion can causes too much distortion and simulation stops.
Typical problems are two, can be seen in (Figure 7.12), where the same cell is plotted during two nodes motion and colored by its quality. The first one is cell inversion. That happens if one or more cells during motion aren't convex anymore so the calculated volume results negative because of numerics. The second problem is cell distortion. This is similar to previous one but volume is close to zero (never negative). Continuity equation (inside the cell mass must be conserved) calculates extreme higher (or lower)



(a) Normal cell **(b)** Cell distorted **(c)** Cell inverted or concave

Figure 7.12: Examples of cells colored by quality index

pressure or temperature in comparison to all the other cells around.
In the first case the simulation stops at the moment of inversion, in the second case after few time steps the run is stopped because of too high gradients.

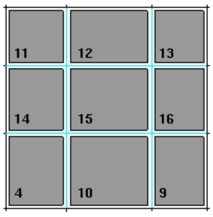
7.2.1 Snapper concept

Snapper routines are software functions to modify the grid during motion. KIVA-4MPI uses only one grid, during boundary motion activates or deactivates cells that are too close to moving surfaces.
Motion method is a simple concept, all nodes defined in moving boundary every time step are analyzed. Piston boundary is always moving, valves one just in a part of cycle. If the boundary moves nodes are moved on the requested direction and quantity. During motion nodes reduce the distance with the other nodes reducing cells dimension and deforming these, when nodes are too close to continue with motion snapper are activated.

Figure 7.13: Simple scheme of snapper

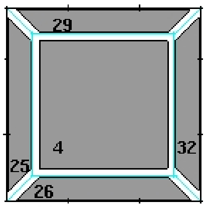
Assuming the example on Figure 7.13 . The original node's coordinate (left Figure) are stored in a table ZSVFACE so during motion (center Figure) the actual coordinate is compared to this value, when the distance between solid node and next fluid node is less than half of original distance, snapper activates (right Figure) and replace moving nodes plane with a new one. The initial moving nodes are placed on original position, faces and nodes boundary conditions are changed from SOLID to FLUID and cell status from FLUID became INACTIVE. Besides old FLUID nodes are converted to SOLID

Table 7.1: Standard blocking scheme and relation table



	12	16	10	14
from 15	top	right	bottom	left
to 15	bottom	left	top	right

Table 7.2: O-grid blocking scheme and relation table



	29	32	26	25
from 4	top	right	bottom	left
to 4	bottom	bottom	bottom	bottom

and MOVING nodes, the original position is stored on ZSVFACE table and coordinates are changed from original to actual moving surface position. Also physical cell and node properties must be transported from old moving surface to the new one, due to continuity, momentum and energy conservation during this transition. The concept is simply explained with following example about cell mass: old fluid cell with some volume bigger that zero is deactivated, inside this was some mass, if this mass just disappear it isn't physical and Navier-Stokes equations have really problem to be solved with continuity. It could happen that the calculated pressure and temperature are lower than before. Also if the simulation keeps running the results would be far away from reality.

Important for those routines is the neighbor table that indicates, for every direction, node next to selected one.

The connection table is the basis to keep connected nodes also after motion and deactivation. It's built on the original mesh when piston is on BDC and valves in closed position. One of the biggest difference and problematic aspect between two presented mesh generation methods is this table. If o-grid is adopted this table cannot be defined and just one direction has this table: the motion direction.

The motivation of this could be find in a simple example: in standard blocking could define relations between cells easily, as shown in Table 7.1 all cells are connected and relations are equivalent in both direction from and to the center cells. In case of o-grid definition in Table 7.2 corner cells (11,13,4,9 in previous example) don't exist anymore, that requires cells around the center to have all relation bottom with the center, but middle cell must have only one top. With this requirement it results to be impossible to write a functional table.

Valve closing and opening are functions included in snapper routines, those change face boundary condition of the cell between valve top and cylinder head. If there is only one layer of cells between two surfaces the moving routine changes into no fluid boundary condition through valve cavity. On the first valve motion instant snapper routine make the first snap, boundary condition are changed and there is no obstacle for the fluid.

7.2.2 Rezoning

As already said previously the standard KIVA-4MPI method for grid moving is the only motion of interested surfaces, all other cells stay still. But there are other type of treatments. For example if no valves are included in the geometry one can adopt pure Lagrangian treatment, that means cells are all moving and squish volumes have the same number of cells from BDC to TDC without any snap during computation. Other treatments are mixed, that means an algorithm is apply to have the best cell quality in every piston position but it's impossible to keep the same amount of cells during the displacement, so also snapper are activated and reduce total amount of cells.
In KIVA-4MPI there are 3 different type of this mixed treatments, called rezone and explained in [2] and [1], those could be used in different grid topologies: "rezcomb", used for asymmetric pent-roof geometries, in which valves are canted with quite different angles between intake and exhaust sides; "rezpent", used for nearly symmetric pentroof combustion chamber; "rezwedge", used for 2-valve engines with wedge combustion chamber.
All included treatments need neighborhood table in all direction because also direction normal to piston and valve motions are interested on this remapping. As already said for new mesh generation method it's impossible to create this kind of table, so those kind of treatments weren't used during this work.

7.2.3 Motion treatment improvement

Presented grid generation method doesn't write neighborhood table and LES requires the big amount of consecutive cycles. Motion treatment need to be improved inside the code, with solution that guarantee in every simulation moment all LES request. Those are: good cell quality, no big jump of volumes and fixed cells topology on same piston position between cycles.

Valve snapper

As already explained above valve snapper routine activates when valve surface is enough close to the next layer of cells, but this position is with relative reference calculated. In every snap cell position will be recalculated, two typical formulation are:

$$z_{ref} = \frac{1}{2}(z_{surf} + z_{ref+1}) \tag{7.1}$$

for nodes above valves, and:

$$z_{ref} = \frac{1}{2}s_{valve} + z_{surf} \tag{7.2}$$

for nodes below valves.
Whereas z_{ref} is the coordinate of nodes that need to be moved, z_{surf} is the surface position,

z_{ref+1} is the neighbor nodes position and s_{valve} is the valve thickness.
Snapper is activated in case of downward moving if::

$$z_{surf} > z_{ref-1} + \frac{1}{2}(z_{surf} + z_{ref+1}) \tag{7.3}$$

in case the valve is traveling upward if:

$$z_{surf} < z_{ref-1} - \frac{1}{2}(z_{surf} + z_{ref+1}) \tag{7.4}$$

for downward travel. z_{ref-1} is the original node position before last snap.
If all these terms were independent from cycle number there will be no problem on the resolution, but snapper is activated if the position of the surface is ahead a certain point. Because of dynamic time discretization this point is variable in every cycle creating fluctuation on z_{ref} and of course on the derivate z_{ref+1} and z_{ref-1} that are calculated with same equation but in a different moments of the simulation.
At first sight this fluctuation seems not influencing a lot nodes positioning during cycles, but thinking the snapper is activated on the same node four times per cycle, one could understand that this fluctuation could become important. Figure 7.14 shows how big this fluctuation in a multi-cycle calculation can be, represented are four slice of four different cycles on the same CA position calculated on one simulation. Those are plotted with velocity magnitude.Fluctuation could be recognized first of all above top valve surface, but also below the same valve. During this simulation not just SGS fluctuations are calculated but implicitly also grid fluctuations are included in the solution.

Solution to this problem was found creating a table, that make a direct link between valve and nodes position.
The first step was creating the table in original grid before every motion begin. The routine calculate automatically how many snaps must be done during complete cycle, than compile a table that assign for every snap threshold an index. Afterward using neighbor relationship a second table, connecting snap index and nodes position for every layer, is compiled. Therefor from actual valve lift the code can calculate how many snaps were already done, where the next snap must be done and all the coordinates needed to nodes repositioning.
Now snapper is activated in case valve is traveling upward if:

$$z_{surf} > z(nsnap) \tag{7.5}$$

for downward travel:

$$z_{surf} < z(nsnap) \tag{7.6}$$

without any cycle dependency.
The repositioning equations were changed in:

$$z_{ref} = \frac{1}{2}(z_{surf} + z_{ref+1}) \tag{7.7}$$

for nodes above valves, and:

$$z_{ref} = z(nsnap, nodes, surface) \tag{7.8}$$

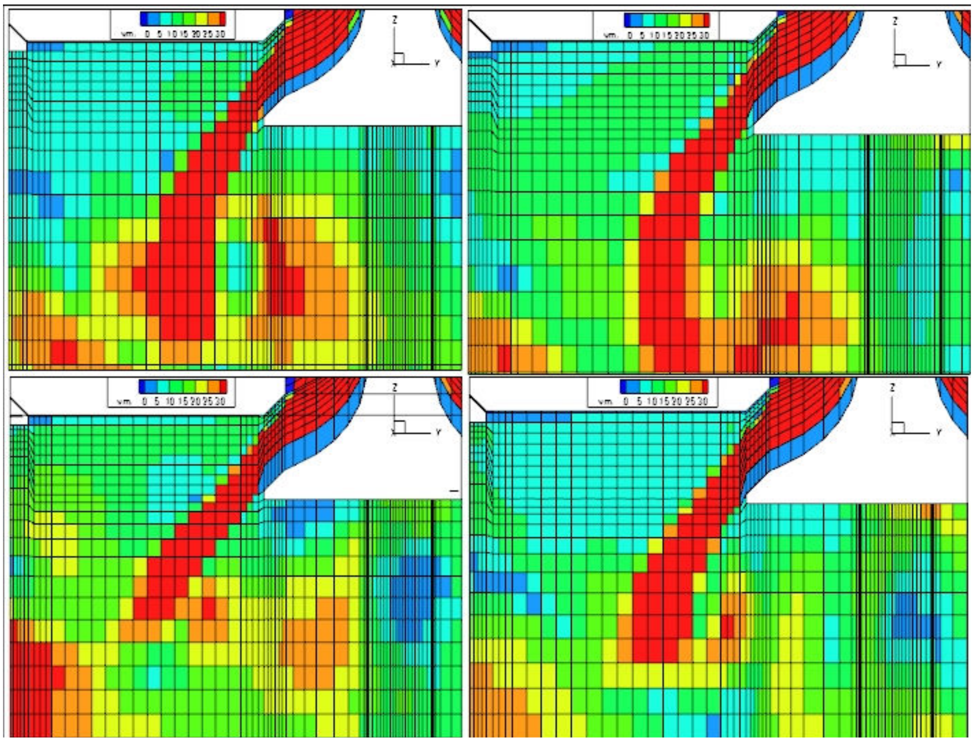


Figure 7.14: Example of grid fluctuation 100° ATDCE colored by velocity magnitude

for nodes below valves.
The first relation is the same as before, but in this case z_{surf} and z_{ref+1} they have less degree of freedom because snapper activation has no more fluctuation caused from z_{ref+1} and z_{ref-1} and the second relation has a fixed reference that never fluctuate.
Also with little degree of freedom in Equation 7.13 as shown in Figure 7.15, representing same CA position in different cycles of one simulation, the issue was solved. The simulation has no fluctuation component due to grid fluctuation anymore.

Piston snapper

Issues for piston motion are similar to the one described for the valve motion. In that case grid fluctuation wasn't so remarkable as before, but could also be a problem in multi-cycle simulation.
Formulation for piston snap was really similar to the valve's one. Those were:

$$z_{ref} = \frac{1}{2}(z_{pist} + z_{ref+1}) \tag{7.9}$$

in case piston is traveling downward, and:

$$z_{ref} = z_{saved} \tag{7.10}$$

when piston travel upward.
Whereas: z_{ref} is the coordinate of nodes that become squish or ghost nodes, z_{pist} is actual piston position, z_{ref+1} is the node above reference and z_{saved} is original moving node

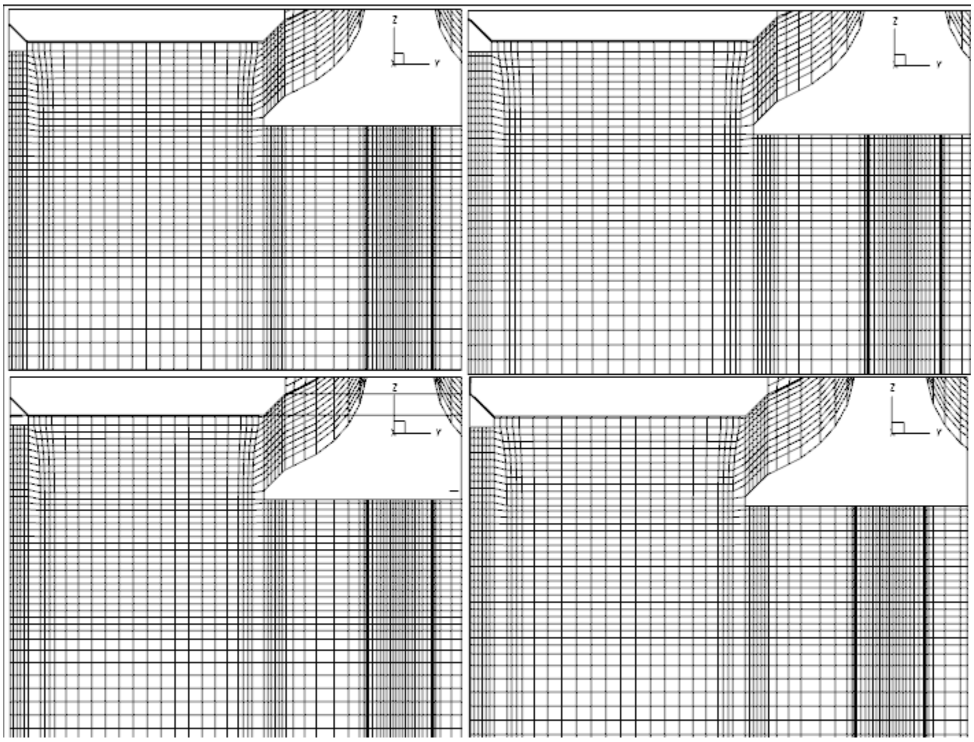


Figure 7.15: Same grid as above after improvement 100° ATDCE

position, saved on previous snap.
Snapper is activated if:

$$z_{pist} > z_{saved} + \frac{1}{2}(z_{surf} + z_{ref+1}) \tag{7.11}$$

in case piston is traveling upward:

$$z_{pist} < z_{saved} - \frac{1}{2}(z_{surf} + z_{ref+1}) \tag{7.12}$$

for downward travel.
For piston motion was necessary to build two tables, one for snapper activation and a second one for correct nodes positioning. Nodes coordinates are really important if the piston travel is long enough to use nodes that also valves use. If piston set wrong coordinate there is a risk of inversion or bad quality cell during computation.
New equations are similar to valve's one.
Snapper activation:

$$z_{pist} > z(nsnap) \tag{7.13}$$

in case piston is traveling upward:

$$z_{pist} < z(nsnap) \tag{7.14}$$

for downward travel.
Therefore positioning equations were changed in:

$$z_{ref} = \frac{1}{2}(z_{pist} + z_{ref+1}) \tag{7.15}$$

for deactivating nodes, and:

$$z_{ref} = z(nsnap, nodes, surface) \tag{7.16}$$

for squish becoming nodes.

Rezoning

The original idea for this improvement was to keep a good grid quality during cycles, especially when the piston is near top death center.
Of course there is no issue for parallel valves engines where all the layer are parallel to each other and no distortion in moving direction appears. Rezoning improvement is needed in case of canted valves engines. With standard treatment when one cell needs to be snapped all the layer follow it. Near TDC it happens something like what is shown in Figure 7.16. Because the same amount of vertical cell is requested in every cylinder section, although the average thickness is smaller as the center one, side cells layer is lower as the center one. This result in a layer of cells quite bigger that the others. This is bad for the equations resolution, because of the volume jump and of course the poor resolution near those wall faces.

Without piston snapper improvement an other problem was that snapped cells into piston cannot come back to the original position.

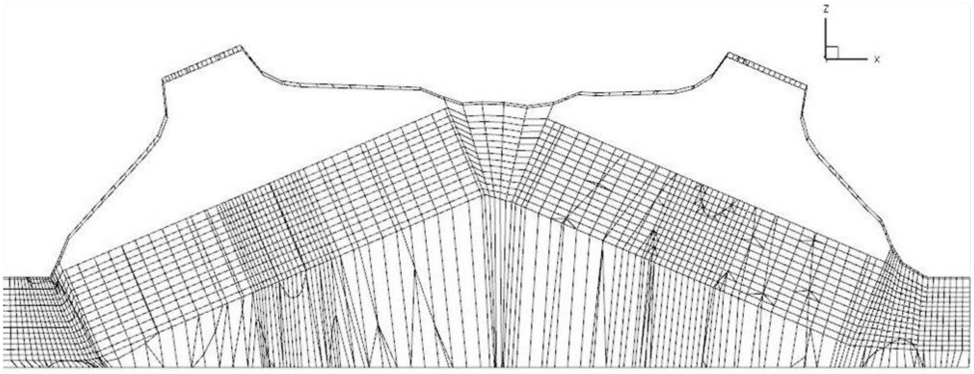


Figure 7.16: Cell distortion on TDC with standard treatment

The idea of a new algorithm was taken from KIVA-4MPI original rezoning but modified to neglect the use of neighbor table, this consist on working only with a limited number of squish layers, defined in input files, set all this cell with the same thickness. Neglect neighbor table means that other parameter to define a common layer must be found. A table with useful properties was created for piston snapper, but wasn't enough because in this case many layers must be analyzed and every cell has different thickness. This issue was solved just creating another subtable every time step, starting from the original one created for the whole domain.
The results are shown in Figure 7.17.

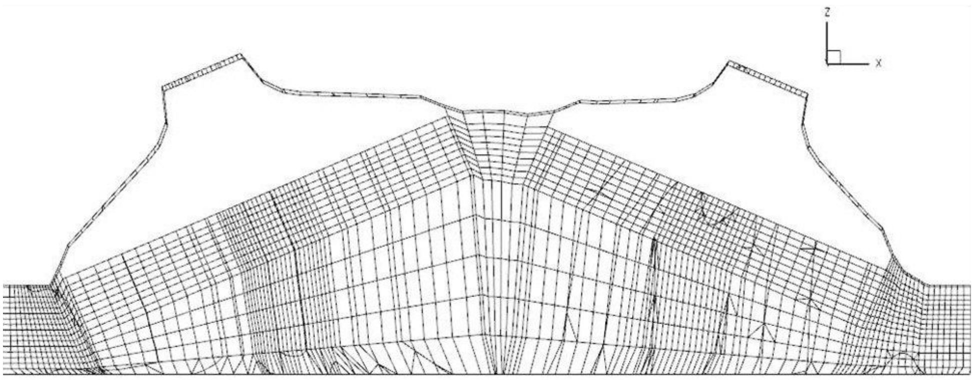


Figure 7.17: Cell distortion on TDC with modified treatment

Wall layer on piston

Included in new rezoning routine there is also an option to keep a certain number of layers with fixed thickness near the piston. That could be useful in the future for near wall flow simulation or flame-wall interaction, where is really important to have accurate solution near wall.
Although this function could be activated without any problem the actual simulations don't need this kind of refined resolution. It was decided to not activate this function in this work.

7.3 LES in engines

Large eddy simulation is a very different method if compared with the most used in the industry RANS, not just for modeling and resolution, but also for the results concept. While in a RANS simulation already single snapshot of a converged solution gives good information about results, LES simulation needs more informations to build a turbulence independent solution.
For a stationary geometry collecting enough results to have good statistic is faster than in a moving geometry where the same boundary condition exist once every cycle, that change the order magnitude of computational costs. Often to reach compromise one collects less results than in fixed geometries.
Code validation in those cases is also really complex, because there aren't simple test cases to use. A geometry with just one valve and moving piston results in a complex study. Validation of the code is done with a stationary and good documented test case. In other cases like in this work experimental data of same geometries are available to working with a validation, also in this case the comparison is quite tricky. During experiments more than thousands cycles could be collected have a good statistic. On the other hand a really expensive simulation could run more than fifty cycles that makes weak comparison in terms of code validation.
All presented cases are the results of a developing method of simulation that needs to be validated and with actual code status and available resources it's hard to say how long it'll take till the method could be use as routine.

Chapter 8

TCC-GM engine

In this chapter the first calculated engine will be described. This configuration was already calculated and good documented experimental data. This simulation will test how good is the code coupled with new grid generation method and implemented models. Particle image velocimetry (PIV) data exist in two positions and measured cycles are more than 3000. Of course it is theoretically wrong compare two data set collecting sample size of different order of mgnitude. Different sample sizes lead to different convergences, as shown by Baum and Peterson [3]. Inside the cylinder some low fluctuation areas need just few cycle to reach convergence, but other high fluctuation zones need more than hundreds cycles to converge. While LES and experiment comparison there are two choices, comparing a converged experimental results of many thousands cycles with a probably not completely converged LES simulation, or compare a similar set of cycles for both. The author choses the second option to have a similarity of convergence during the comparison, but also complete converged experiment will be considered.

8.1 Configuration

Geometry and boundary condition were reproduced from the experimental setup presented by Sick [35], trying to reach most precise results as possible.

The engine geometry is rather simple. Two parallel valves with flat head and piston. Intake and exhaust ports are composed by two curves and finish into two big tanks, built to give a stable pressure at the port boundary.

Easy to understand that those huge tanks have a big impact on simulation computational cost, therefor first simplification is done neglecting those two tanks and appling the measured pressures at the boundary. Those shows tank pressure is not affected by big fluctuations on different cycles, see Figure 8.2. For that reason tanks were excluded from the computational domain and instead of a constant pressure on the boundary condition inside tanks, variable pressure was applied with crank angle. Applied pressure distribution is the same shown in Figure 8.2.

Other geometry data are resumed in Table 8.1.

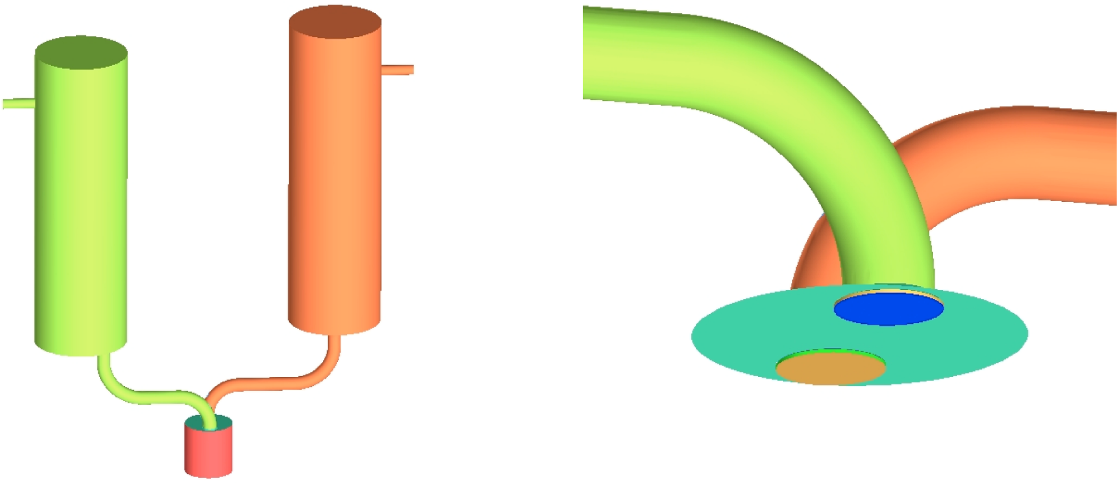
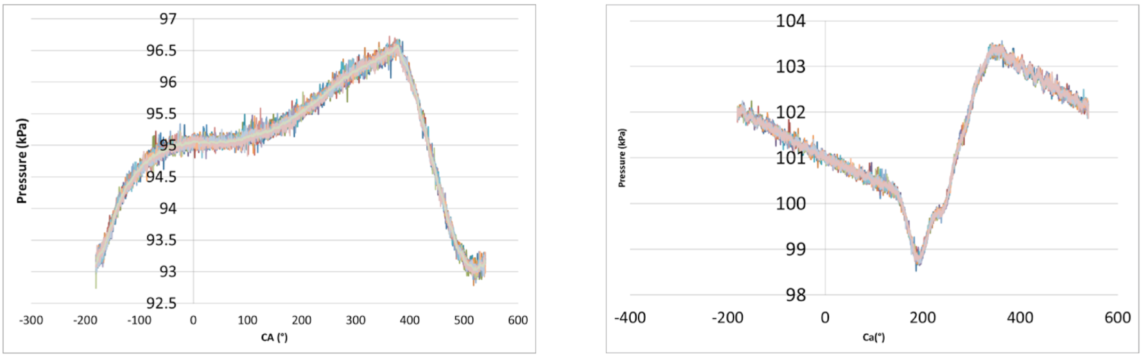


Figure 8.1: Geometry used in the experiment

Table 8.1: Basic engine data

Cylinder bore	$92mm$
Piston stroke	$86mm$
Conrod length	$234.96mm$
Swept volume	$571.7cm^3$
TDC volume	$63.53cm^3$
Compression ratio	$1:10$
Engine speed	$800rpm$
Operation temperature	$318K$



(a) Intake pressure

(b) Exhaust pressure

Figure 8.2: Pressure fluctuations on the boundaries: experimental data

8.2 Results

First comparisons that every simulation needs first are zero dimensional (0-D). Those comparisons are needed for a first validation and setup calibration. Sometimes although geometry and boundary conditions perfectly match with experimental setup, first results do not match. This kind of validation is very quick, few cycles are needed, two or three are often sufficient. Pressure and mass fluctuations in non-fired cases are limited between cycles because those values are already averaged in the whole cylinder volume.
In case of differences between experiments and calculation some changes will be applied on the setup. Typical issue that often occurs is the too high TDC pressure on the compression stroke. This difference is caused by geometry inaccuracy, in-cylinder mass variation or inaccuracy on pressure boundary condition.
Geometry inaccuracies are typical in case the simulation is done before measurements and during experimental setup building theoretical quote are not perfectly matching. It is always better verify stroke and clearance to have same TDC and BDC volumes to neglect this kind of issues.
In-cylinder mass variation can be a consequence of valve motion, volumes or density errors. As already explained in Chapter 7, valves opening and closing transition are not the same as the real case. If the initial space between valve and cylinder head is too great, the simulation does not let same amount of air inside the cylinder causing a lack of mass during compression phase. Volume errors can only be caused on geometry, as explained before. Density errors can be consequences of pressure or temperature issues. If simulated wall temperature is too difference than the real also fluid is influenced, so it is important to apply correct boundary condition on cylinder walls and also fluid temperature on boundaries.
Last considered issue is the boundary condition pressure accuracy, simulation considers a constant pressure at the whole surface that can not correspond to the real case. Furthermore, simulation is not considering local pressure loss due to diameter decreasing or other kind of obstacle that in experimental setup are present. In actual case pressure are measured on the interface between tanks and port, if simulation apply the measured pressure it is impossible to consider loss created by the strong diameter reduction.
Several ways to solve 0-D validation issue are used. Some people change geometry reducing stroke or increasing piston-head clearance on TDC position until goal pressure is reached. This variation brings change on the instantaneous piston position, velocity and acceleration those are influencing pressure gradients, if this happen when local velocity are compared, big gaps appear between experiment and simulation. Other solution, chosen in this work, is to reduce slightly the pressure at the boundary condition. Applying a constant pressure on the whole surface little loss that can not be simulated is included. In the present case this reduction is less than 7% on both boundaries. It is important to consider the same percentage reduction in both boundaries, otherwise gradients are changing during intake and exhaust strokes.

Figure 8.3 shows in-cylinder pressure variation on simulated cycles and measured ensemble average values, as before said the simulated cyclic variation of this value is negligible, just $30kPa$ at TDC position. Using boundary condition changes the simulated pressure fits good with experiments that have on TDC position a pressure fluctuation of $10kPa$.

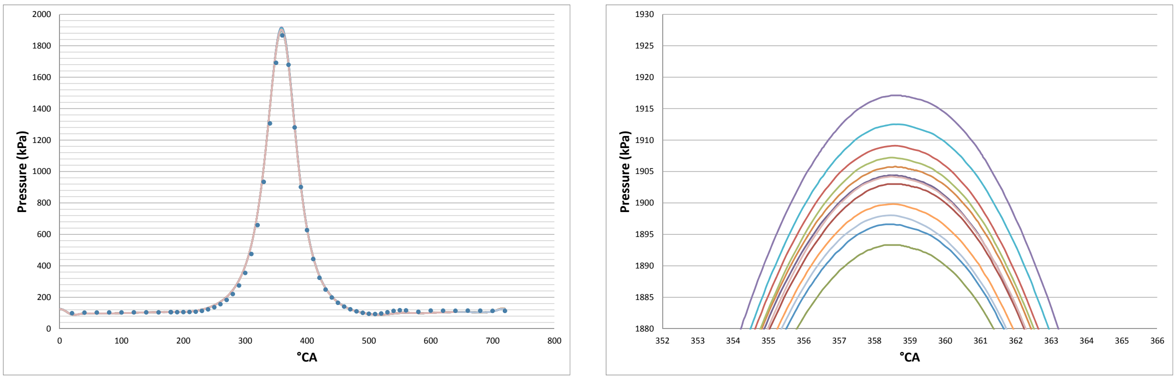


Figure 8.3: Pressure fluctuations in-cylinder: complete cycle and TDC zoom

Figure 8.4 gives information about in-cylinder mass in every piston position and for different cycles. In that case seems that this parameter does not fit with experimental results. To verify the goodness of this parameter in the simulation just compression phase without valve was simulated, the results showed that is impossible to reach the goal pressure with given mass, maybe some misunderstanding on given data occurred.

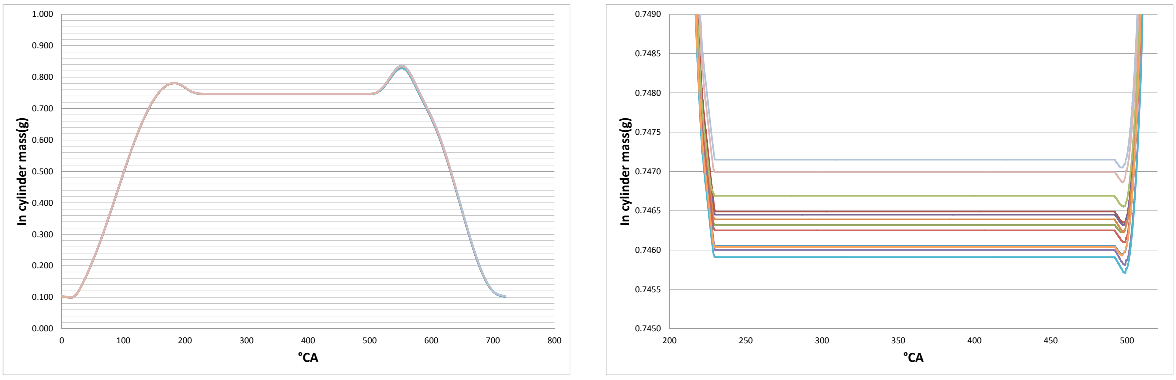


Figure 8.4: Mass fluctuations in-cylinder: complete cycle and compression stroke zoom

Second and more precise kind of validation was done with 2-D data obtained from PIV measurements. This kind of experiment gives local 2-D velocity information in a limited cylinder area. The measurements are done for hundreds or thousands engines cycles to calculate a proper average and root mean square (rms) of velocity fluctuation value in each point of caption grid.

In the present case caption window has dimension 60x50mm and 2mm resolution, its coordinates are showed in Figure 8.5 for position 100CA ATDCE. In the second analyzed position, 300CA ATDCE, the window is reduced to 60x20mm.

Figure 8.5 shows also lines, vertical and horizontal, where comparison is done.

Position 100ATDCE

Considered position is on the middle of intake stroke, the fresh air is entering in cylinder through intake valve and piston velocity is near the maximum. Therefore, flow velocities are really high. In this case valve lift is enough high to have limited flow velocity but some engines have 1-2mm valve lift and much higher velocities.

Below diagrams are disposed as follow: on the left hand side the mean velocity, opposite the fluctuation (rms), on the top there are the highest sections in the geometry, below other sections sorted by z coordinate.

First of all must be said that simulated cycles are just 40 and statistic convergence is not reached. If some discrepancies appear must be considered this point, especially in high fluctuation zone, where convergence is reached after more cycles.

As first example in Figure 8.6 simulated fluctuations are much higher than experiments. Velocity has a good match in second and third diagram, but first and fourth diagrams have not good match. Simulation in the first plot shows a very high velocity in a really thin section, because of the coarse resolution PIV can not capture high gradients like that one. This is demonstrated from fluctuation higher than normal and from Figure 8.10, where the same plot are reported but with experiment averaged on 3000 cycles. Here can be notices how velocity distribution on this first section changes, fluctuation still remain very high but other sections have not important variations neither in velocity nor in fluctuations.

Fourth plot has not clear explanation, the inverse gradient can be caused from few number of simulated cycles.

Figure 8.7 shows w velocity at the same sections analyzed. Also in this case remark are similar as above. In the first plot experiment does not show big gradient , but in Figure 8.11 where the 3000 cycles are plotted, and the grid is finer, bigger gradient appears. Third plot fits quite good but fluctuations remain too high. Other two plots have same velocity magnitude order but the distribution seems moved to the left hand side. Also in this case a bigger data set is required.

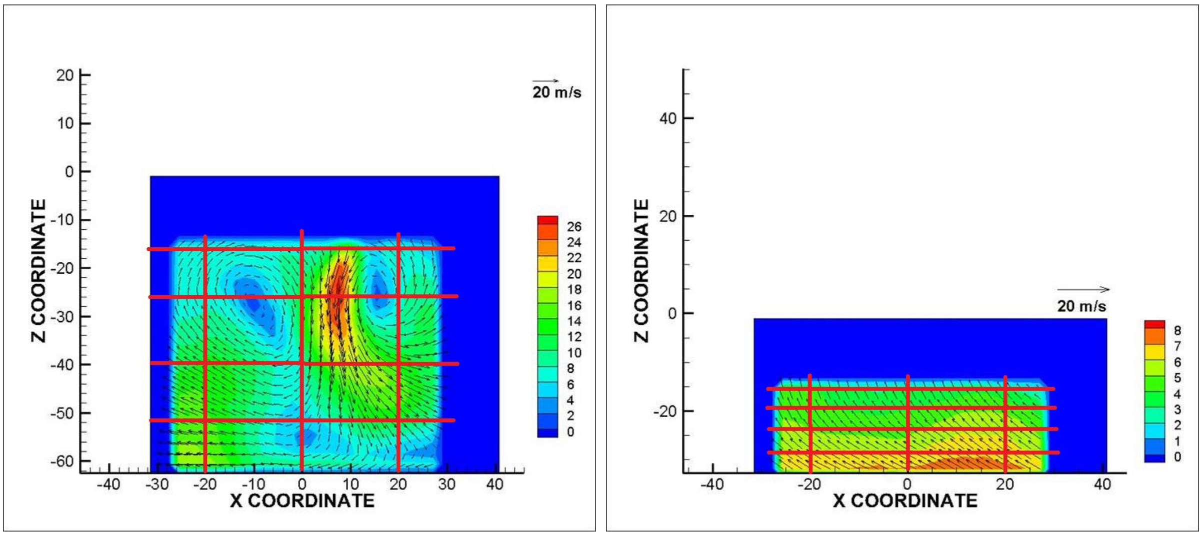


Figure 8.5: PIV caption windows

Three vertical sections are considered, left hand side section is positioned on the figure top, the rest are sorted by x coordinate.
In this case fluctuations are very high and the velocity distribution does not fit proper with experiment in Figure 8.8 but more averaged cycle and refined experiment grid in Figure 8.12 reduce the difference between experiments and simulation. In the experiment of Figure 8.12 was not possible to have the same number of plots because the caption area was reduced so the top section data were not collected.
Some better results can be seen on the velocity w in vertical sections (Figure 8.9), where distribution fit quite good the experiment especially the larger data set in Figure 8.13.

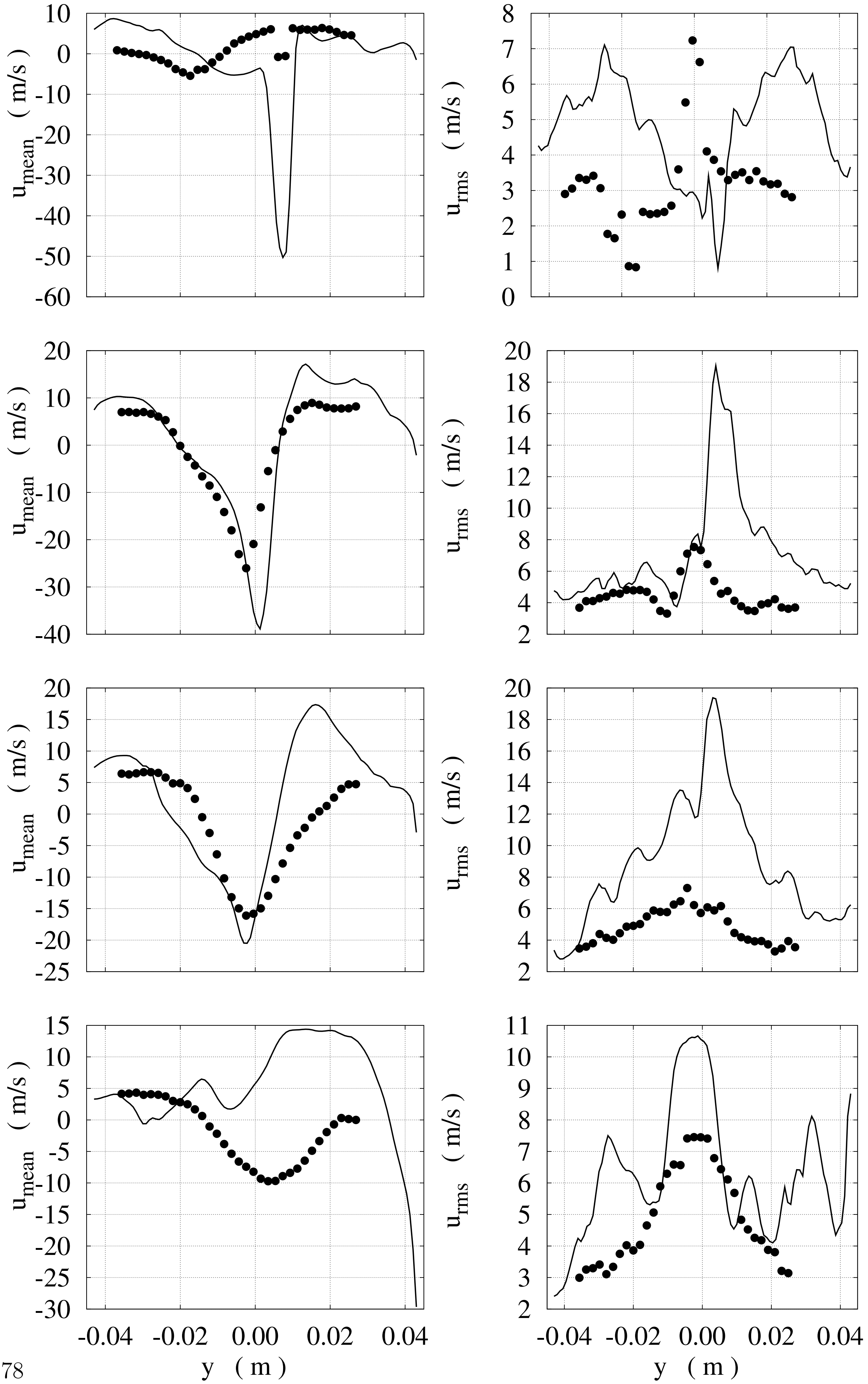


Figure 8.6: Average u velocity and fluctuation for horizontal sections 100ATDCE

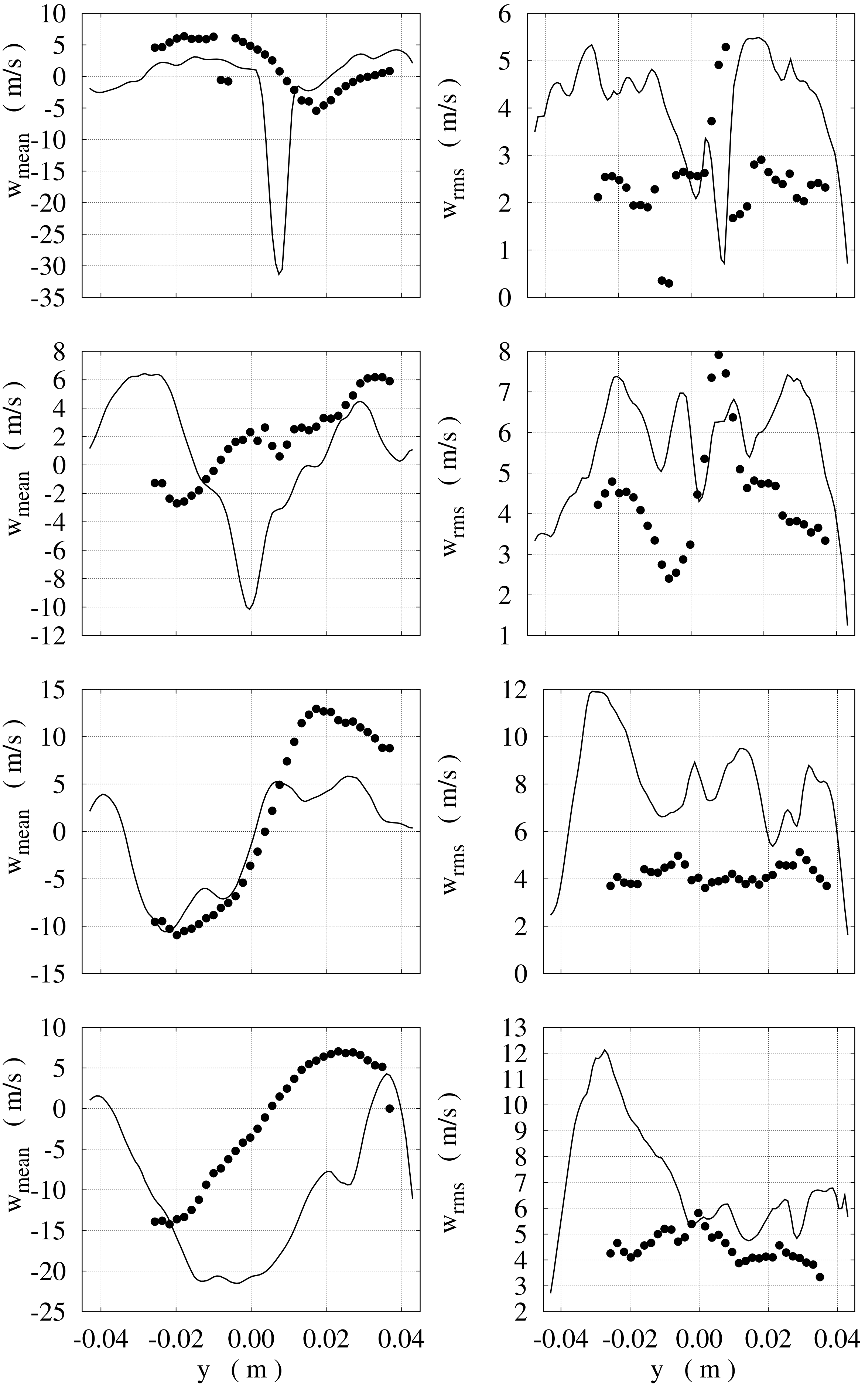


Figure 8.7: Average w velocity and fluctuation for horizontal sections 100ATDCE

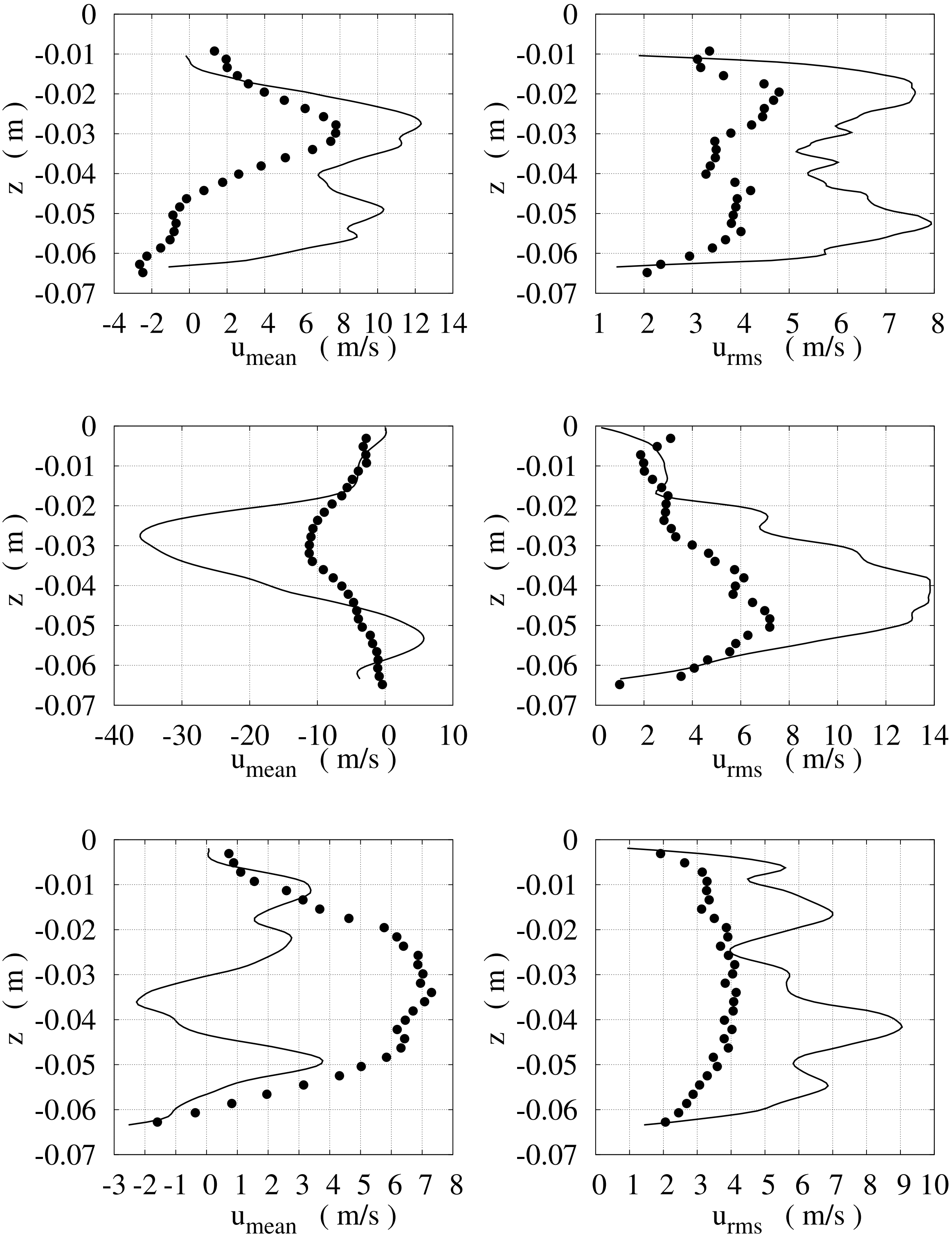


Figure 8.8: Average u velocity and fluctuation for vertical sections 100ATDCE

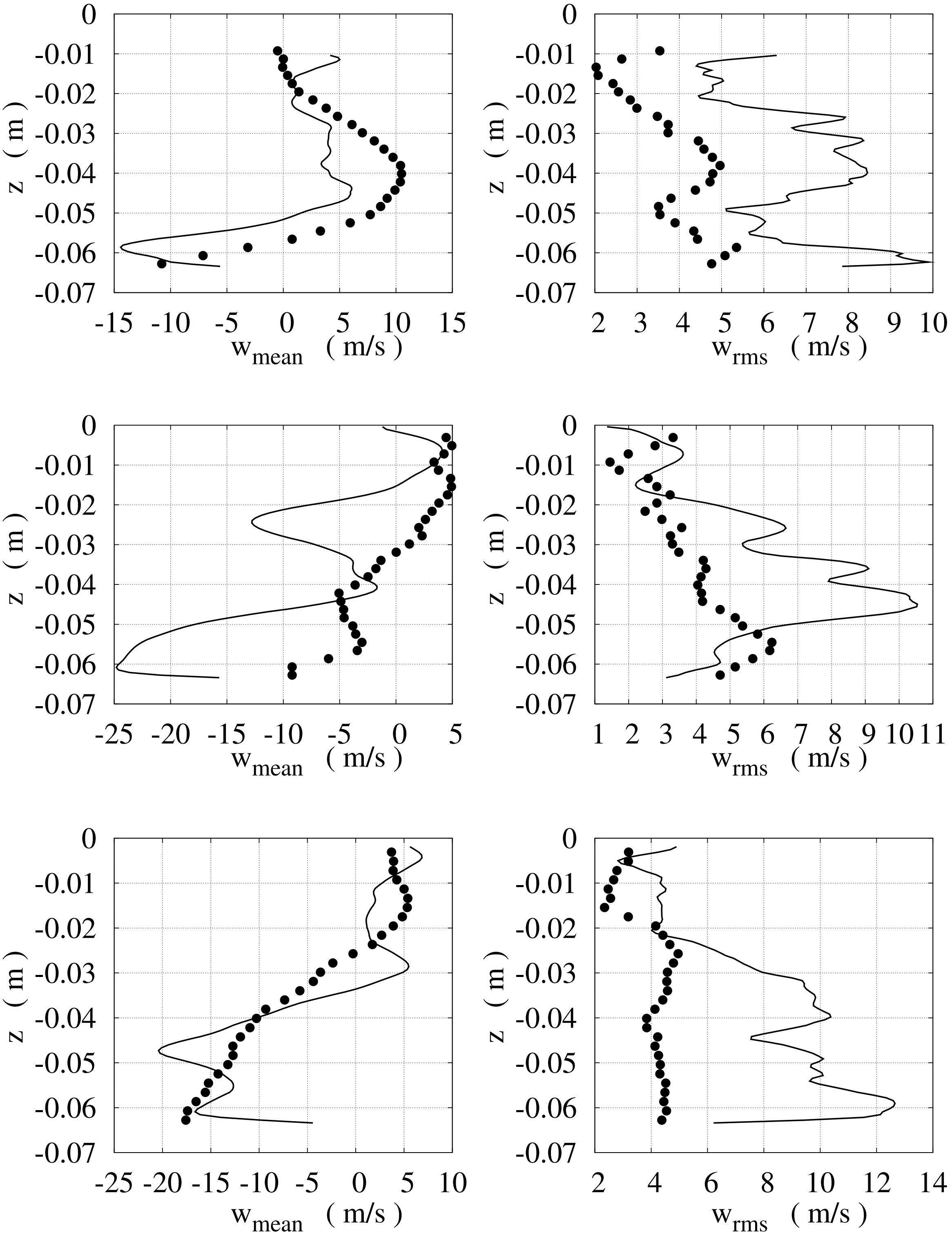


Figure 8.9: Average w velocity and fluctuation for vertical sections 100ATDCE

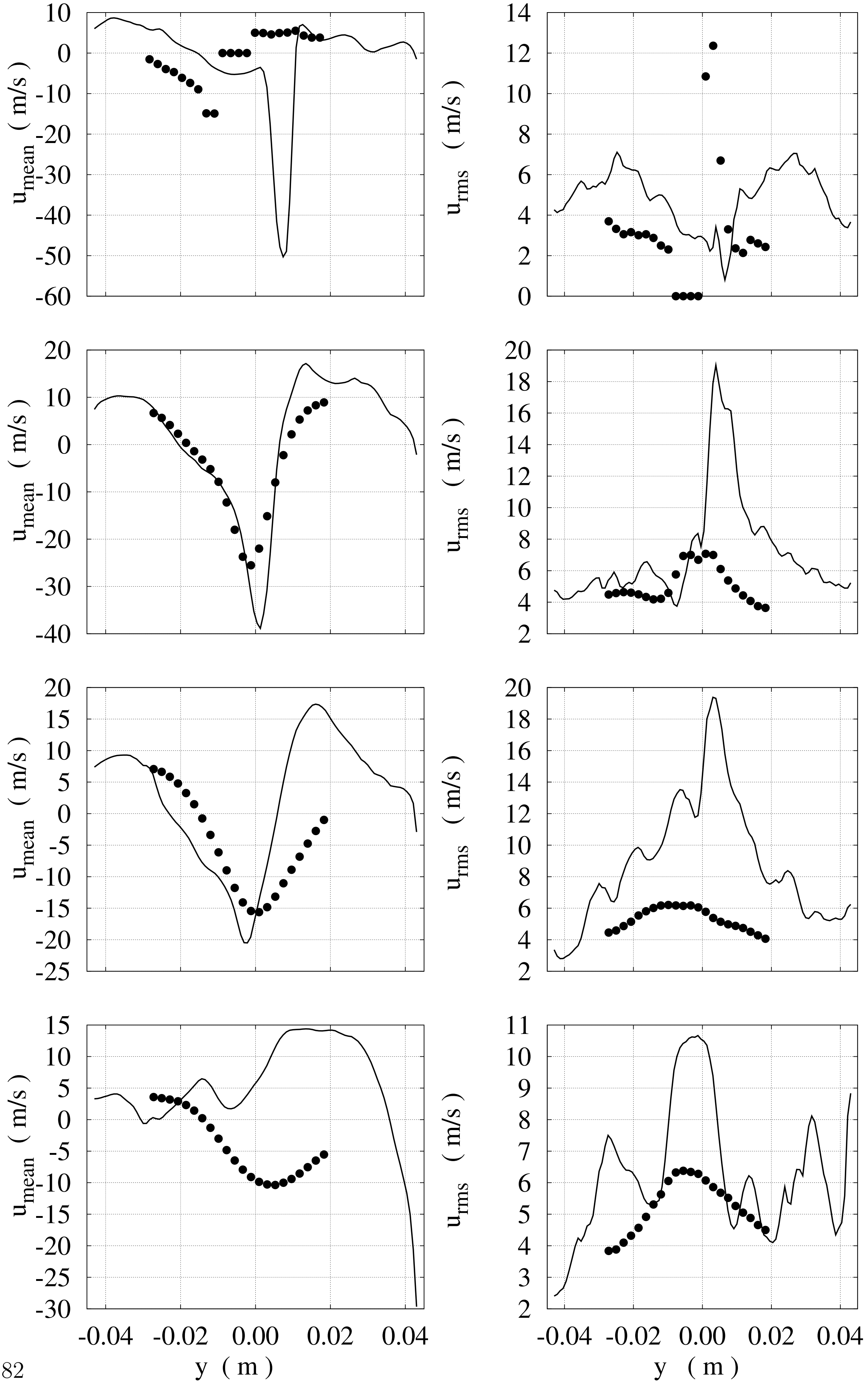


Figure 8.10: Average u velocity and fluctuation for horizontal sections 100ATDCE

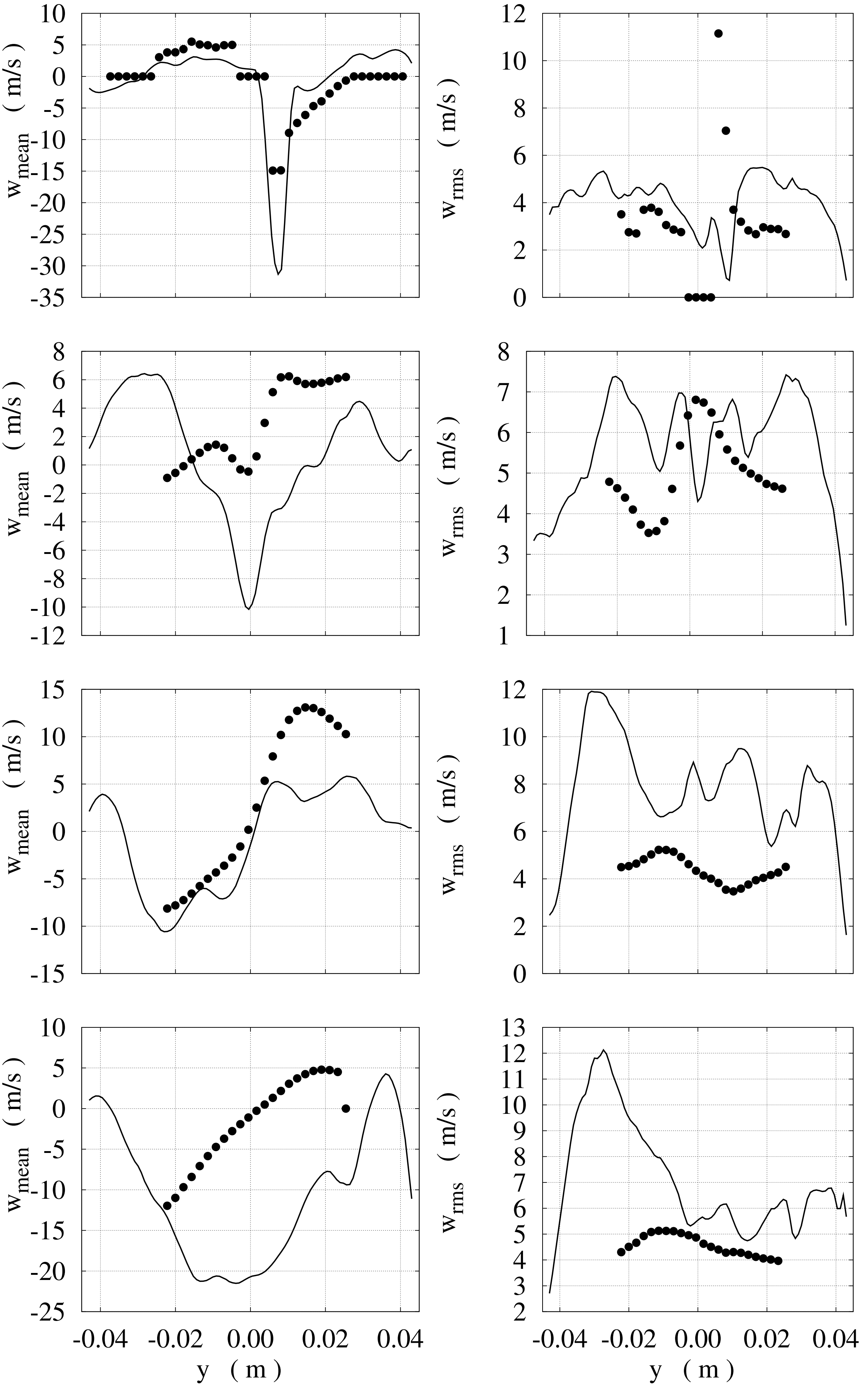


Figure 8.11: Average w velocity and fluctuation for horizontal sections 100ATDCE

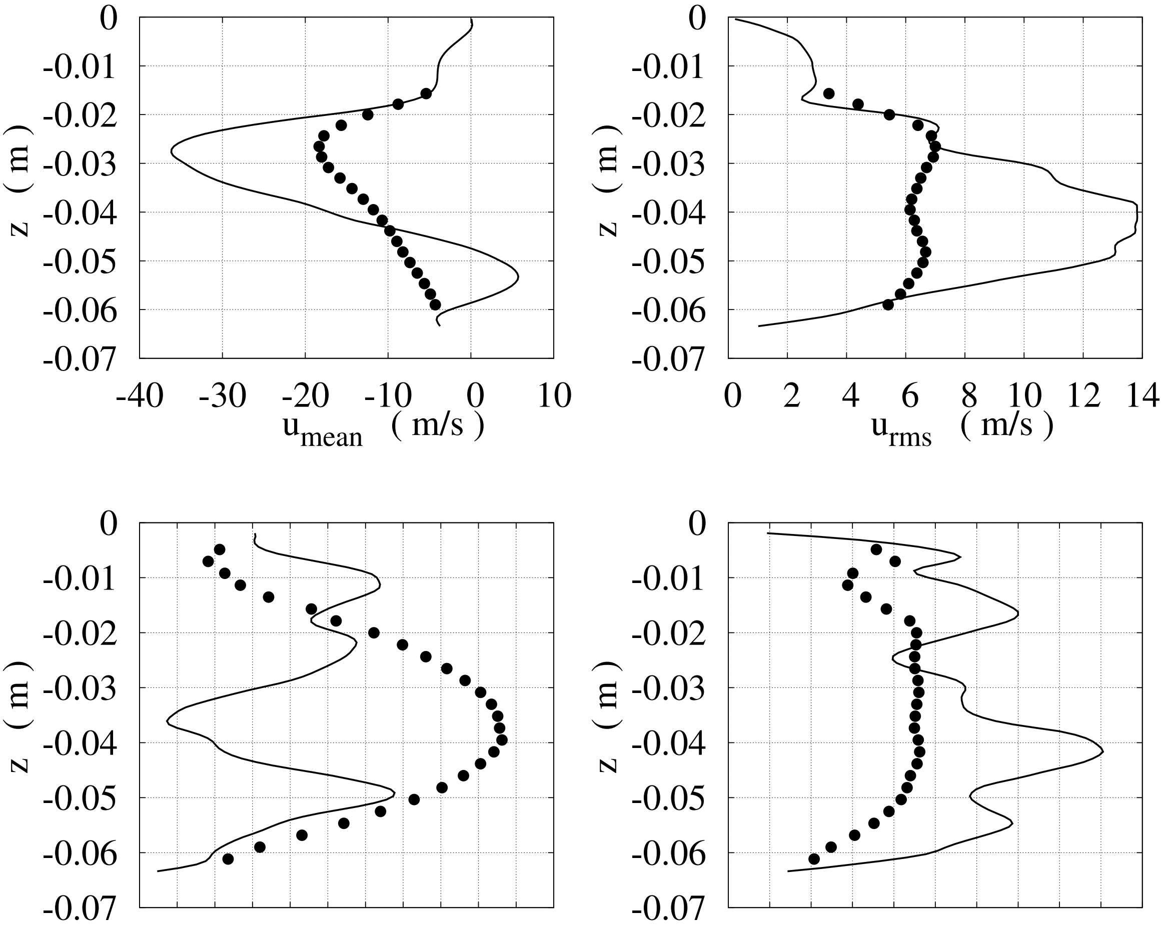


Figure 8.12: Average u velocity and fluctuation for vertical sections 100ATDCE

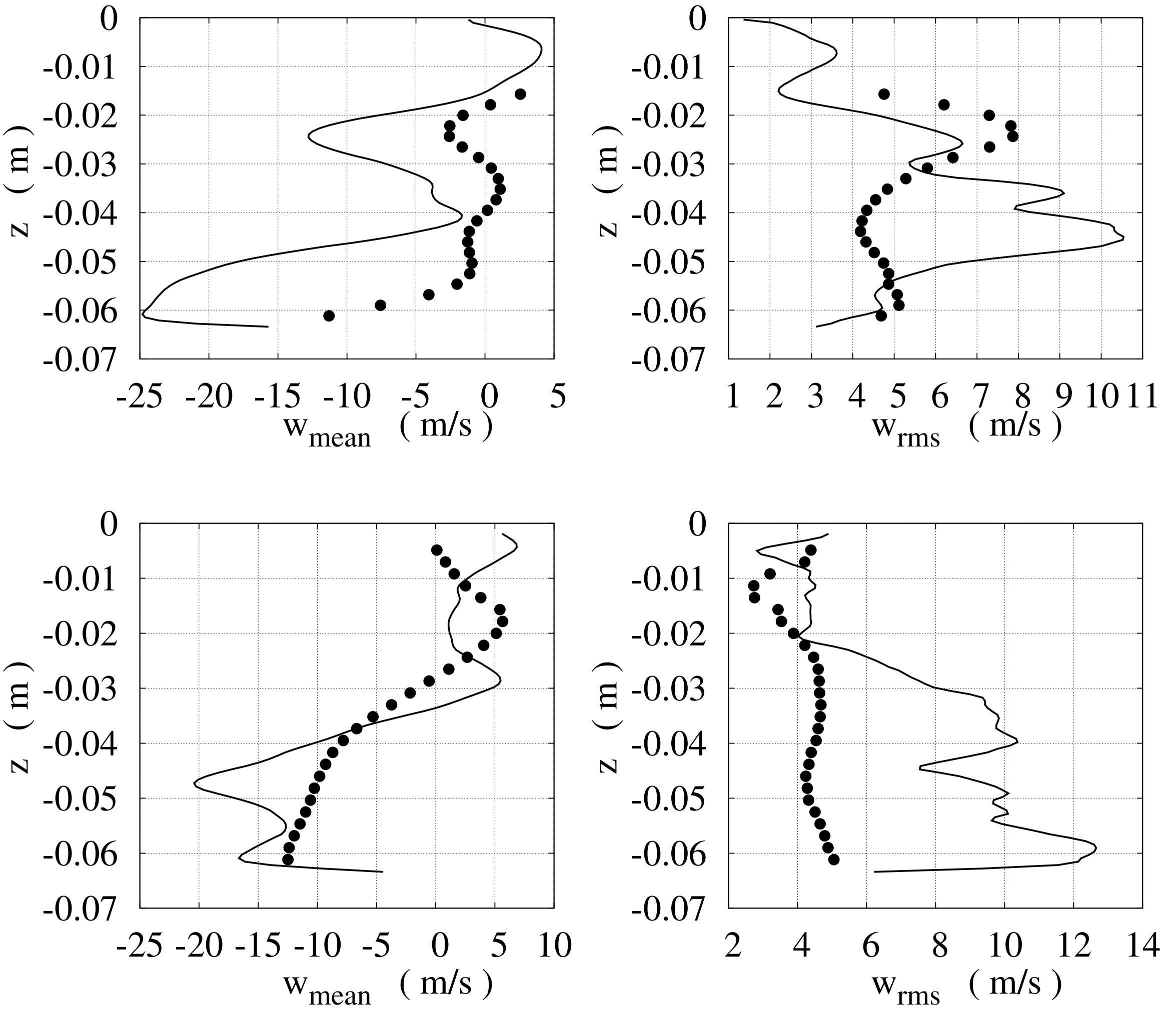


Figure 8.13: Average w velocity and fluctuation for vertical sections 100ATDCE

Position 300ATDCE

This second analysis is done on the compression phase, both valves are closed, piston is near TDC. Therefore velocities and fluctuations are much smaller than previous case. Plots distribution is the same adopted previously.

Plots in Figure 8.6 have a good match with experiment and fluctuations are still higher than experiments. This position requires less cycles than 100ATDCE to reach convergence due to reduced fluctuations.

The only issue can be found in fourth plot where the simulated velocity maximum is moved to the left side compared to experiment. However the fluctuation are higher than on the other side, so the discrepancy happen because of the lack of robust data set.

Figure 8.15 shows u velocity at the horizontal sections. Here there is a good match, except for the second plot near the wall were simulated velocity are much bigger than measurement. Looking at Figure 8.19, where 3000 cycles experiment are reported, velocity and gradient on this area are increasing. So also this case can be explained with difficulties on measurements, but not completely because 3000 analyzed cycles are collected on limited area. For that reason complete informations cannot be collected.

Vertical sections showed in Figure 8.16 have some problem on the highest section, here simulated velocity are quite smaller than experiment results. Other two sections have good agreement.

Figure 8.17 shows plots of z velocity on vertical sections, where simulation fit good in all three analyzed positions.

Because of reduced fluctuations of the case differences between 100 cycles and 3000 cycles data sets are limited.

Figure 8.18, Figure 8.19,Figure 8.20 and Figure 8.21 reports the second available data set.

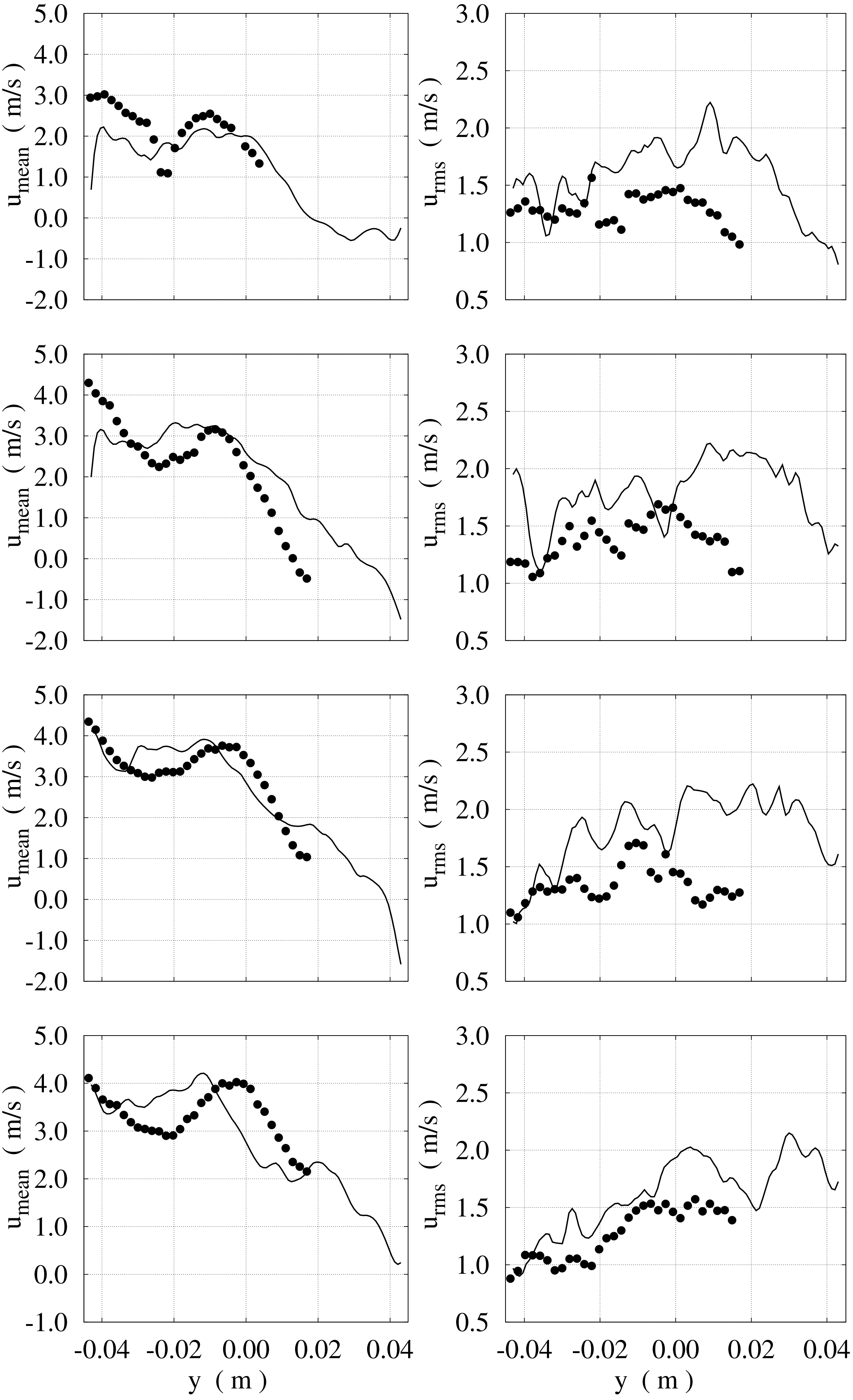


Figure 8.14: Average u velocity and fluctuation for horizontal sections 300ATDCE

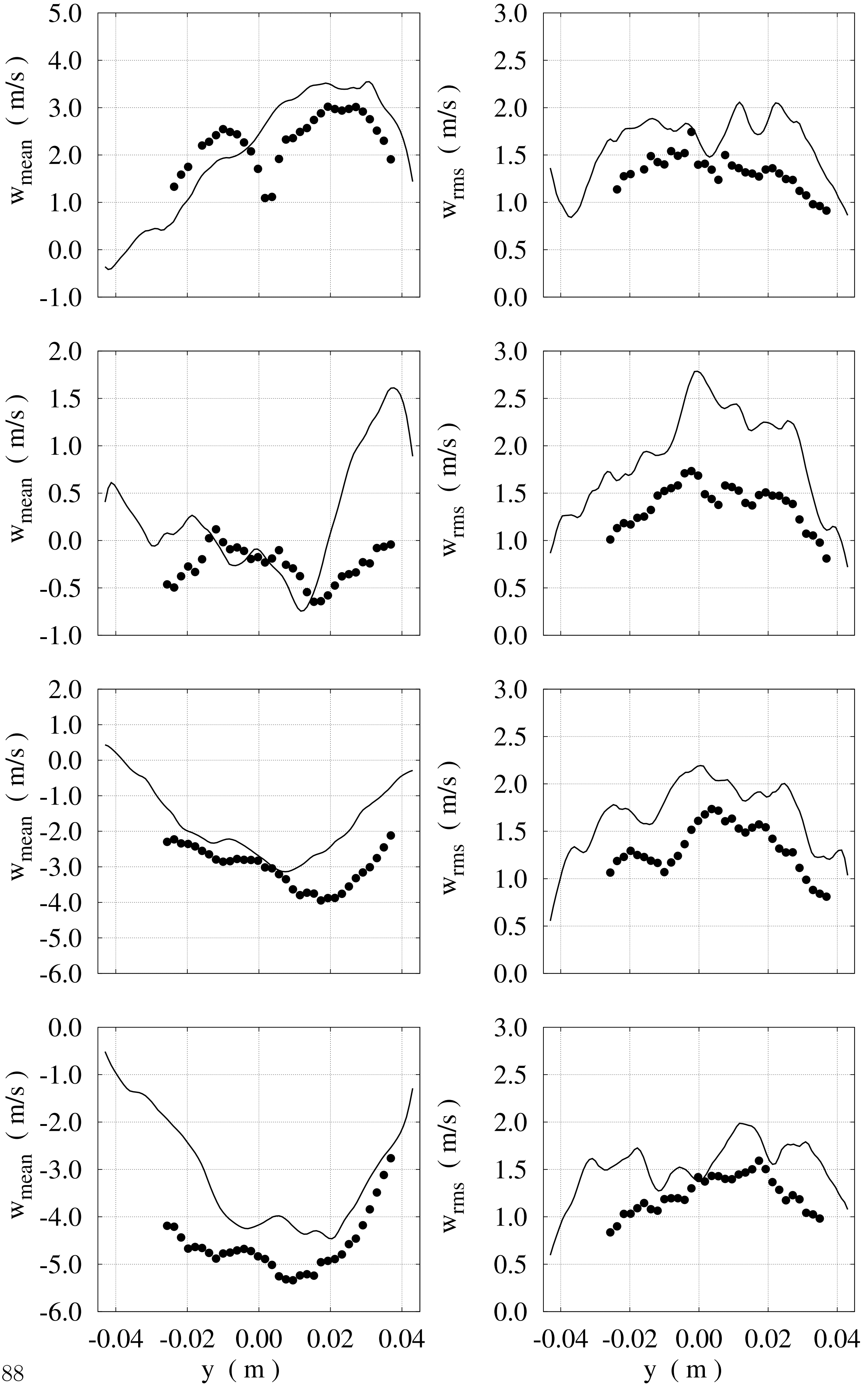


Figure 8.15: Average w velocity and fluctuation for horizontal sections 300ATDCE

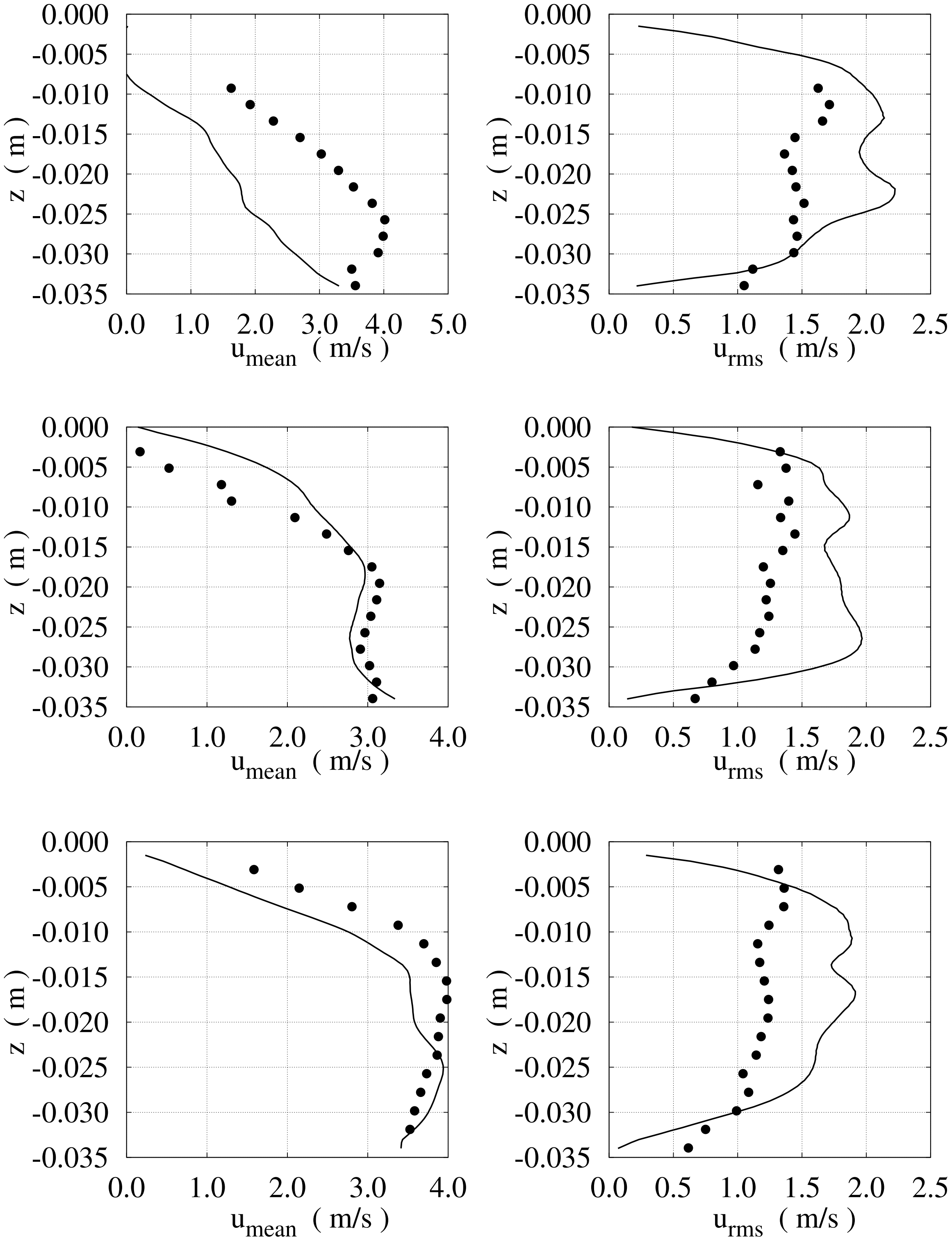


Figure 8.16: Average u velocity and fluctuation for vertical sections 300ATDCE

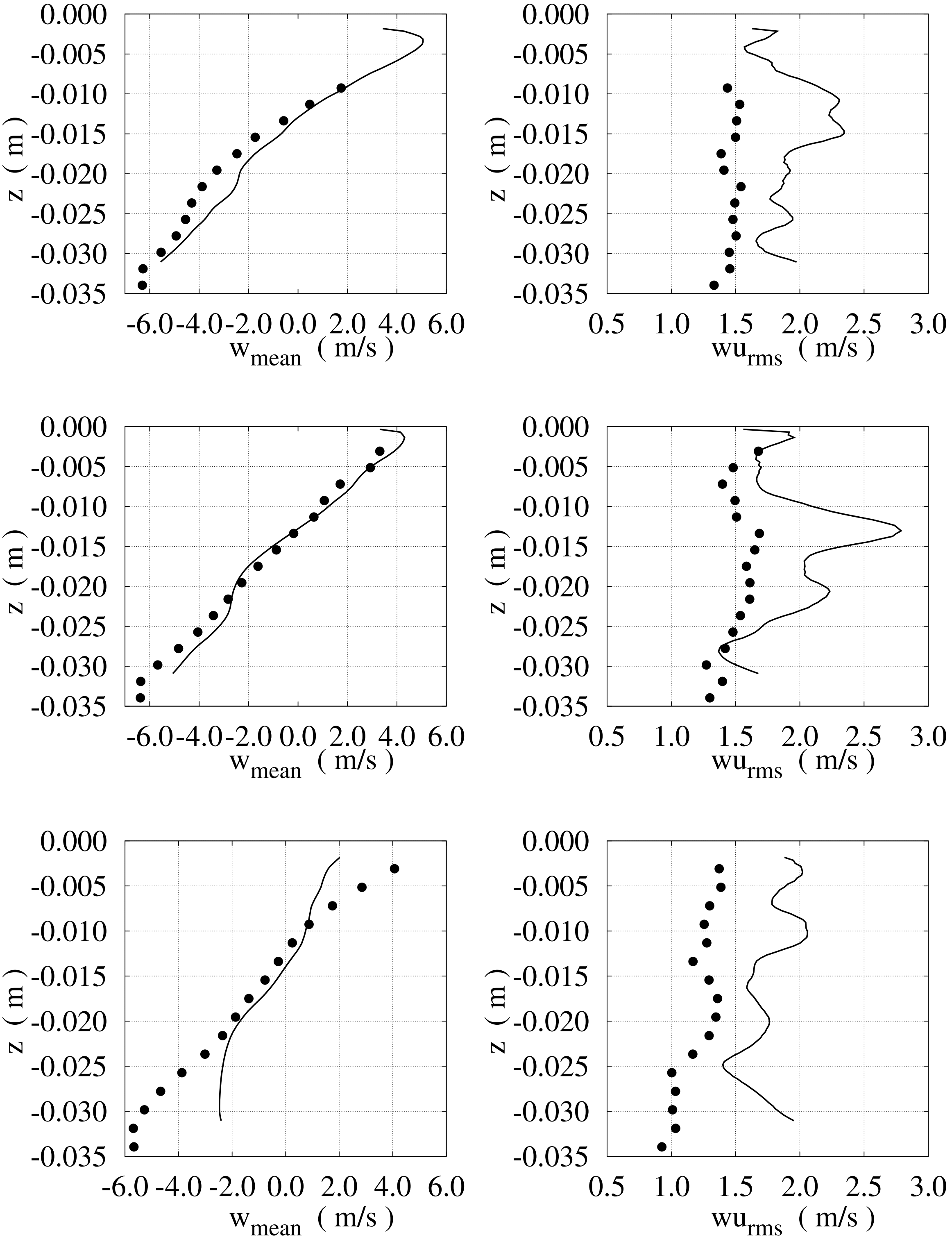


Figure 8.17: Average w velocity and fluctuation for vertical sections 300ATDCE

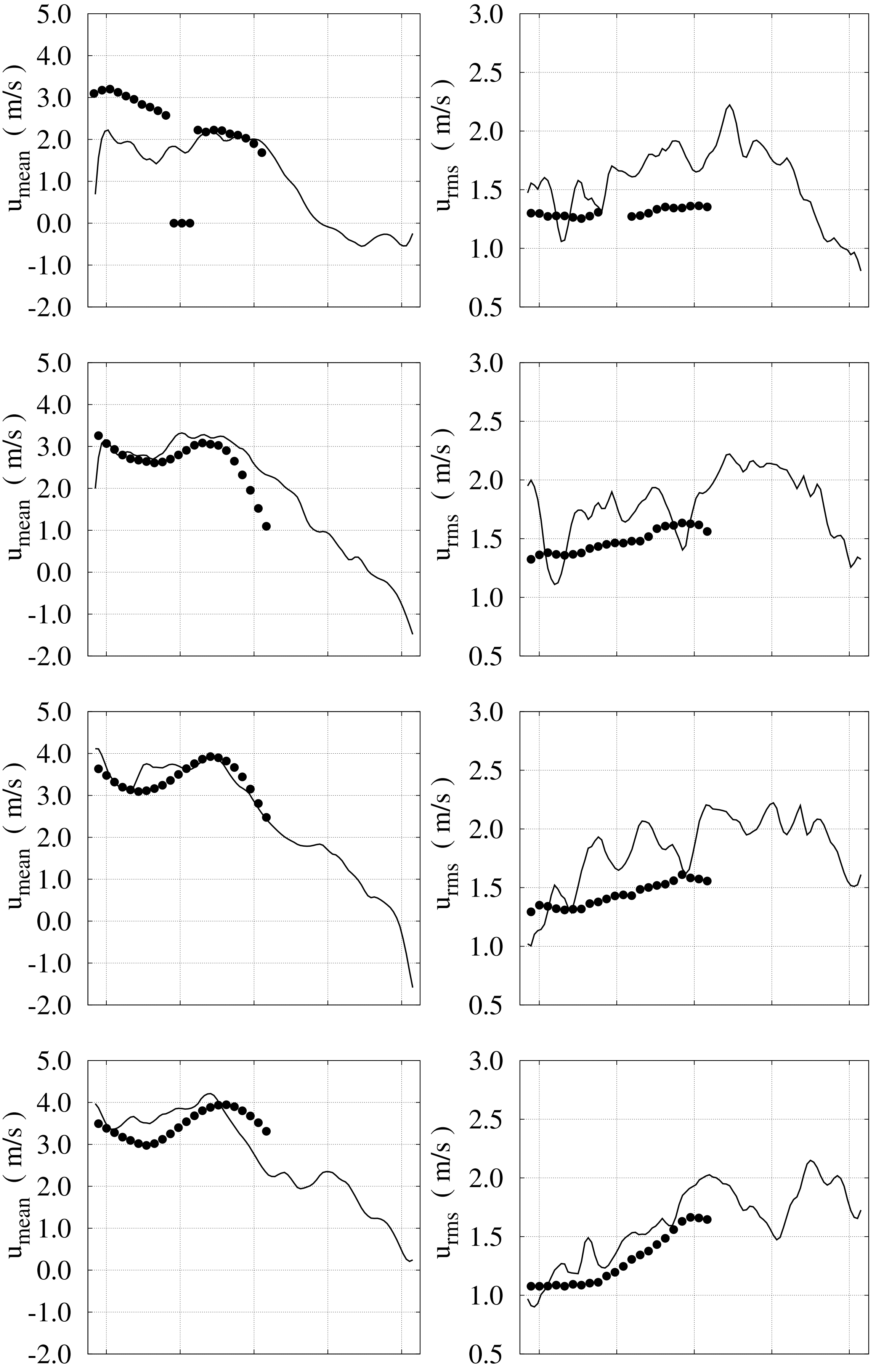


Figure 8.18: Average u velocity and fluctuation for horizontal sections 300ATDCE

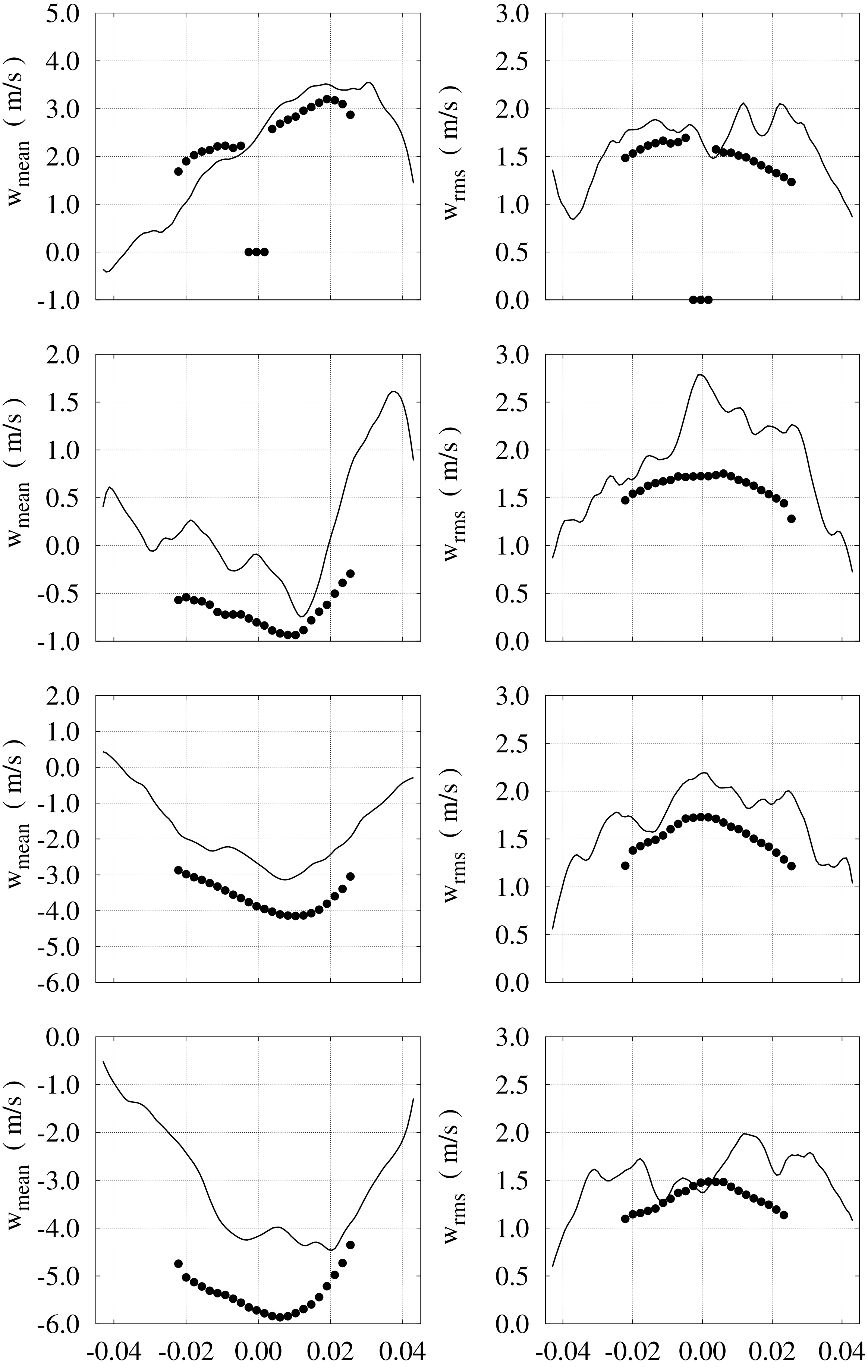


Figure 8.19: Average w velocity and fluctuation for horizontal sections 300ATDCE

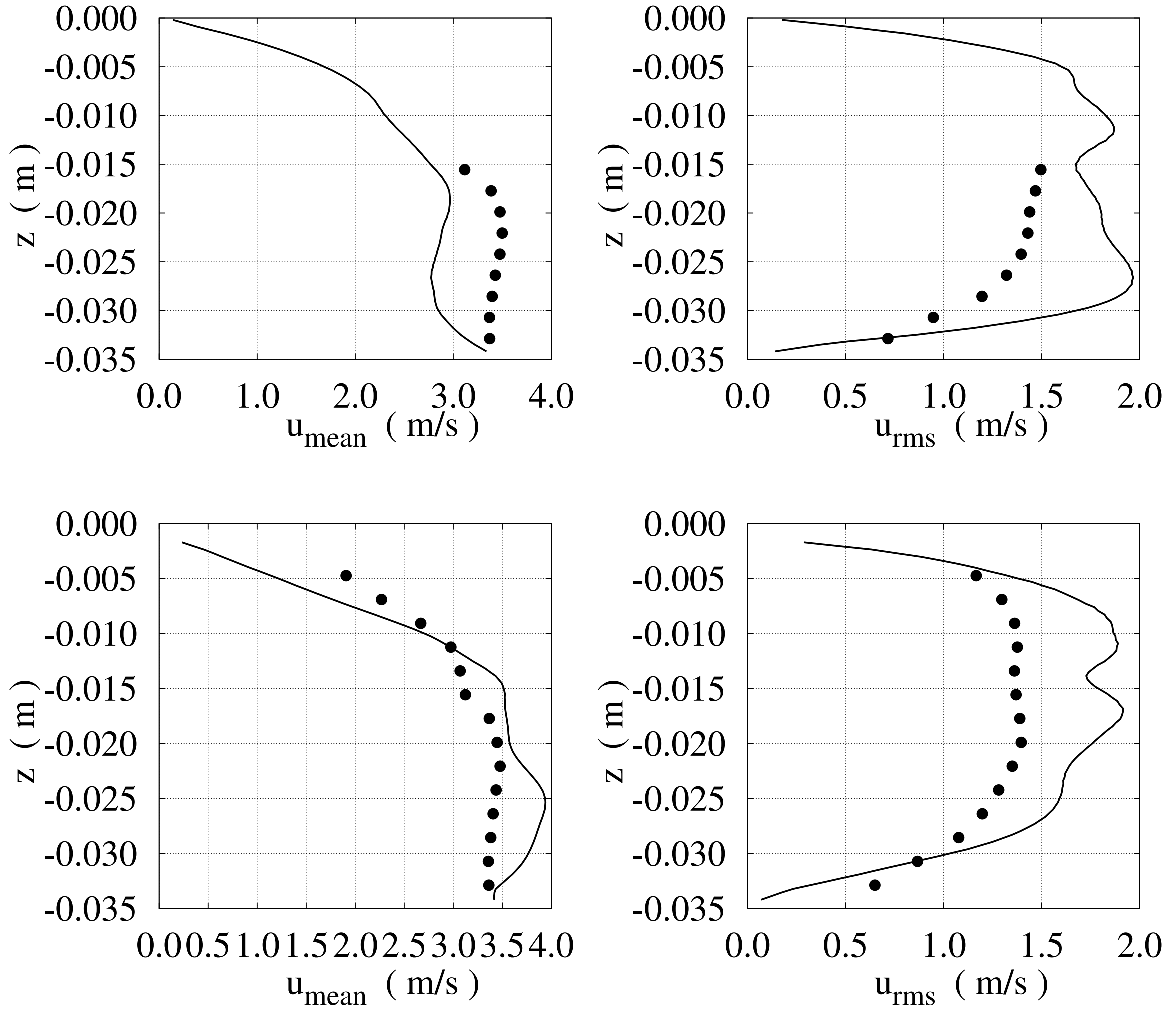


Figure 8.20: Average u velocity and fluctuation for vertical sections 300ATDCE

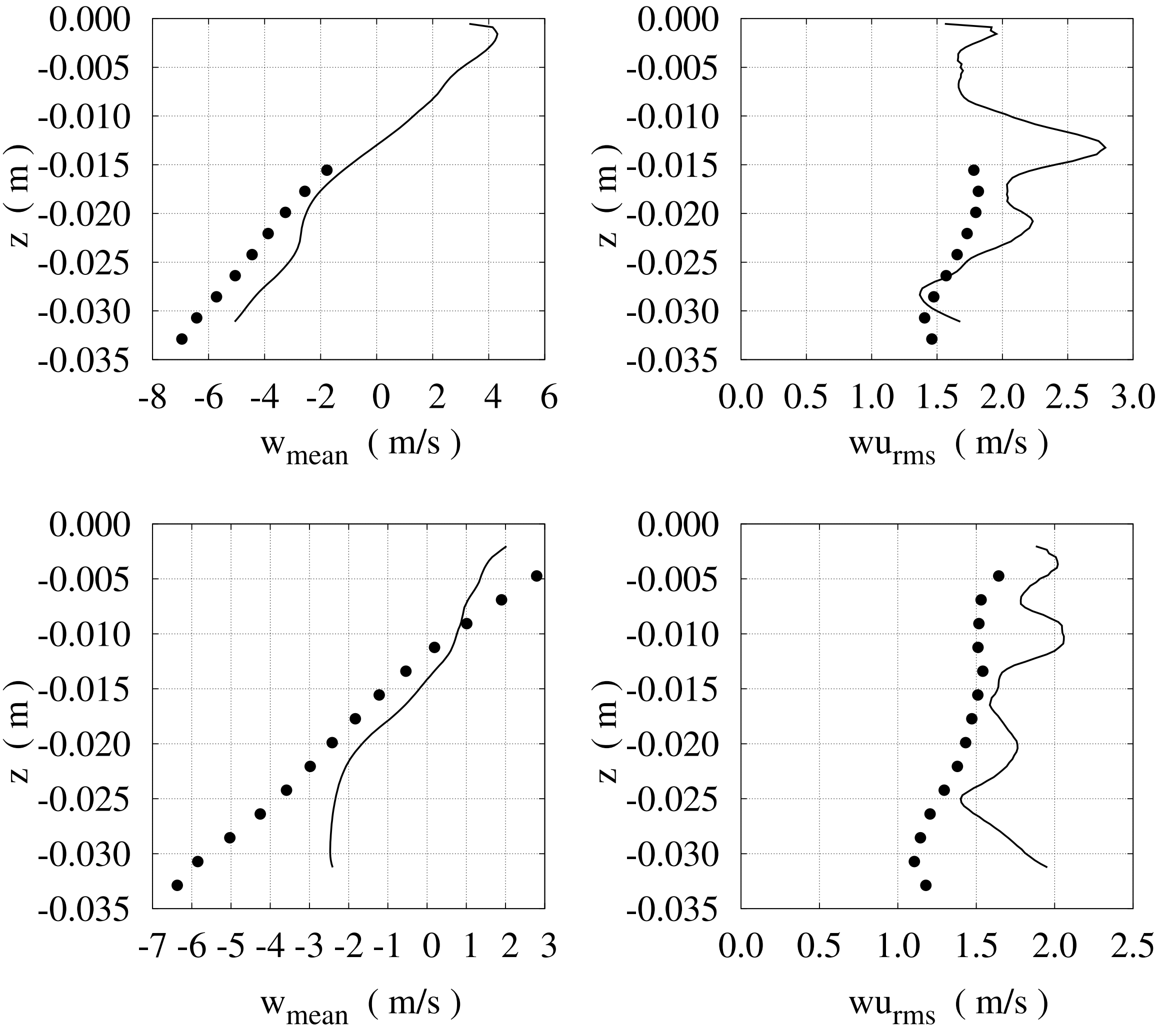


Figure 8.21: Average w velocity and fluctuation for vertical sections 300ATDCE

8.3 Fired case

When this work is done, experimental results of fired case did not exist but they were planned. To have some advantage on the results and understand if the code can run this simulation, some test were run. All used data were extracted from motored case or other simulations, with all the consequences this can have.
First of all gasoline was selected as fuel, its oxidation is simulated with one step chemistry. Other nine secondary reactions are included to stabilize the combustion. Used reactions are:

$$\begin{gathered}
4\,\mathrm{C_8H_{17}} + 49\,\mathrm{O_2} \longrightarrow 32\,\mathrm{CO_2} + 34\,\mathrm{H_2O} \\
\mathrm{O_2} + 2\,\mathrm{N_2} \longrightarrow 2\,\mathrm{N} + 2\,\mathrm{NO} \\
2\,\mathrm{O_2} + \mathrm{N_2} \longrightarrow 2\,\mathrm{O} + 2\,\mathrm{NO} \\
\mathrm{N_2} + 2\,\mathrm{OH} \longrightarrow 2\,\mathrm{H} + 2\,\mathrm{NO} \\
\mathrm{N_2} \longleftrightarrow 2\,\mathrm{N} \\
\mathrm{O_2} \longleftrightarrow 2\,\mathrm{O} \\
\mathrm{H_2} \longleftrightarrow 2\,\mathrm{H} \\
2\,\mathrm{CO_2} \longleftrightarrow 2\,\mathrm{CO} + \mathrm{O_2} \\
2\,\mathrm{OH} \longrightarrow \mathrm{O_2} + \mathrm{H_2} \\
2\,\mathrm{H_2O} \longrightarrow 2\,\mathrm{H_2} + \mathrm{O_2}
\end{gathered}$$

The same scheme is tested with 1-D flame to proof flame thickness and velocity before 3-D engine simulation started.
Simulation is run with air/fuel mixture near stoichiometry as normal for SI engines. The spark ignition is applied on 345ATDCE, 15CA before TDC expansion stroke.
The simulated domain is the same used in motored case and the grid has no changes, afterwards will be clear that finer grid is needed. In the boundary condition wall temperature is changed but pressure on open boundaries is the same as before, due to that an error is introduced.
Initial solution is set with exhaust gas in-cylinder and the start point is TDC intake stroke, so fresh air enters in-cylinder before ignition occurs. Other points were tested but there were always issues with the exhaust stroke. If exhaust valve opens and inside the port fresh mixture is present, combustion starts and propagates in ports creating a lot of instabilities, so was decided to start at the end of exhaust stroke to neglect this issue.
However exhaust stroke has always issues. Due to high in-cylinder temperature and pressure, combined with the small valve lift at the beginning of exhaust stroke, high gradients appear. 30-40CA long velocity near the valve is really high. Due to Crack Nicholson time discretization scheme, which limits time steps with CFL number, the simulation needs much more time in this interval. 30% of computational time every cycle is used for the 50CA after exhaust valve opening where high velocities appear in exhaust port.
Because of long computational time just few cycles run this case. Therefore this simulation is used as test to verify if this kind of physics can be simulated with the CFD

code.

Figure 8.22 shows pressure distribution in seven calculated cycles on two different simulations. As predicted fluctuations are much bigger than motored case. The pressure maximum cyclic fluctuation can reach $200kPa$ excluding the first cycle of every run. There is also a strong gradient on the maximum pressure position of $1-2CA$ and in one case pressure peak is delayed of 10CA. Temperatures have quite big cyclic variation, those are plotted in Figure 8.23.

Although the lack of experimental results, some comments on simulations can be done. First of all, speaking about the pressure can be easy noticed that maximum values are quite high and reach values of $40bar$. Starting from a motored cycle with maximum pressure values around $19bar$ this can be considered on the normal range.

First fired cycle must be discarded from the data set. In this case the initial temperature was wrong, as shown in Figure 8.23 while the setup was around $300K$ after second cycle temperature on initial position $0CAATDCE$ is stabilized around $800K$. Of course the wrong initialization influences in-cylinder mass, as shown in Figure 8.24, on those cycles and increases pressure peak.

Strange phenomena can be noticed on the right hand side of Figure 8.23, where all curves around position $359CA$ have a temperature jump. This can only be some artificial effect created by a numerical failure. Looking deep into the simulation was noticed that this jump happens in correspondence of the last piston snap (see 7.2.1). In this phase with strong temperature gradient a snap can cause damages. As shown by Figure 8.25 the same space occupied by two layer of cells on the left figure, few time step later is occupied by one layer. As long one cell have no wall boundary condition its temperature is conditioned by neighbor cells through energy balance equation. On the other side if the cell has one surface as wall defined its temperature is strong conditioned by the wall. In this case a important part of volume inside the domain is influenced directly by wall temperature and the consequence is wrong calculation of the whole volume mean temperature. Possible solution for this issue can be a thin wall cell set, at least two or three in z direction, so the volume influenced from the wall does not have a strong variation. A routine which apply this modification was already implemented by the author, but was not applied in this simulation.

Last result showed on this simulation are flame position data, extrapolated from temperature or fuel mixture fraction, Figure 8.26 shows average fuel mixture $10CA$ after ignition point and Figure 8.27 uses average temperature to determine flame position.

Temperature fluctuations of 6 cycles are showed in Figure 8.28, fluctuations in the flame position are visible and one can see how the combustion is more advanced in some cycles than others.

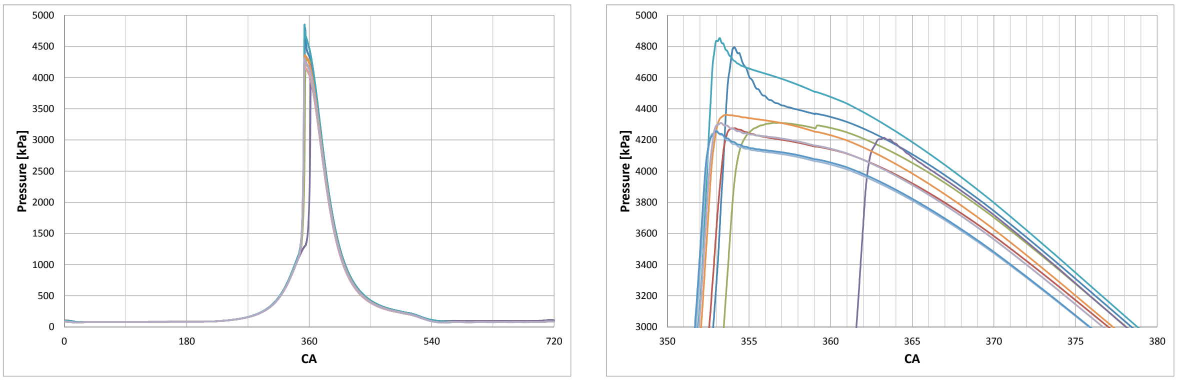


Figure 8.22: Average in-cylinder pressure with cyclic variation

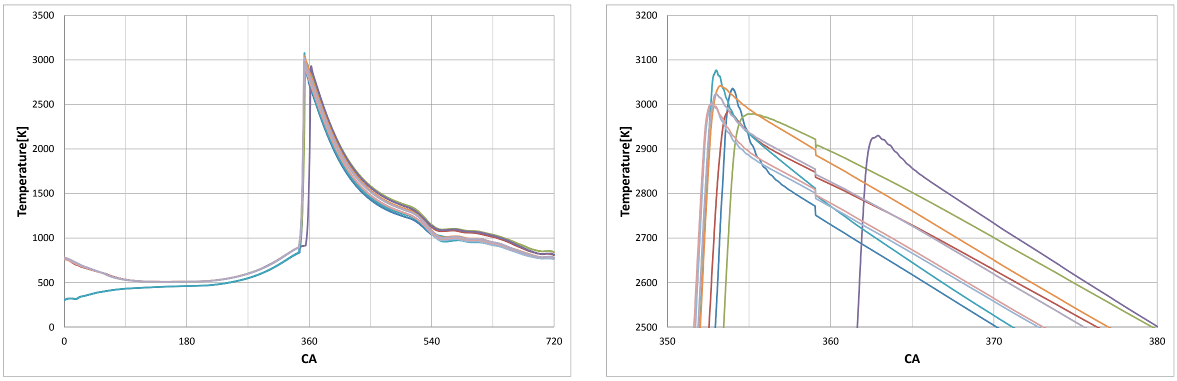


Figure 8.23: Average in-cylinder temperature with cyclic variation

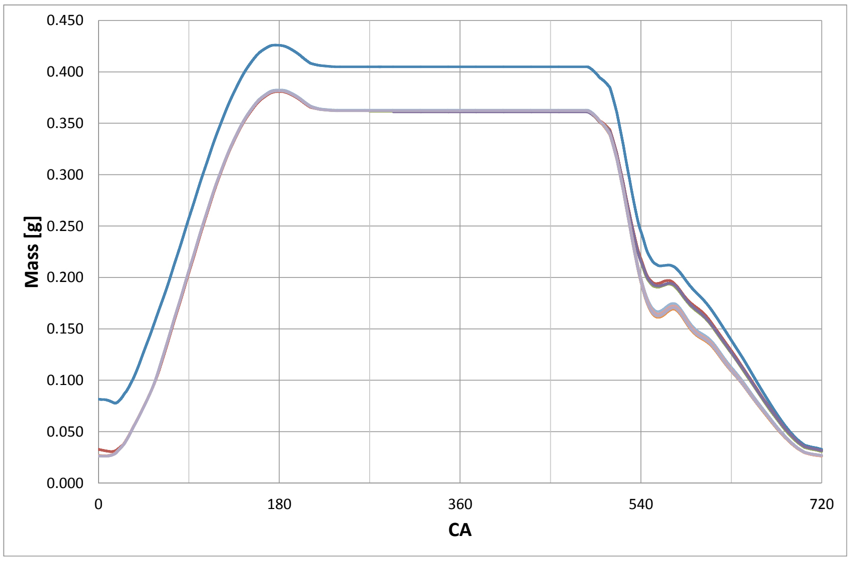


Figure 8.24: Average in-cylinder mass with cyclic variation

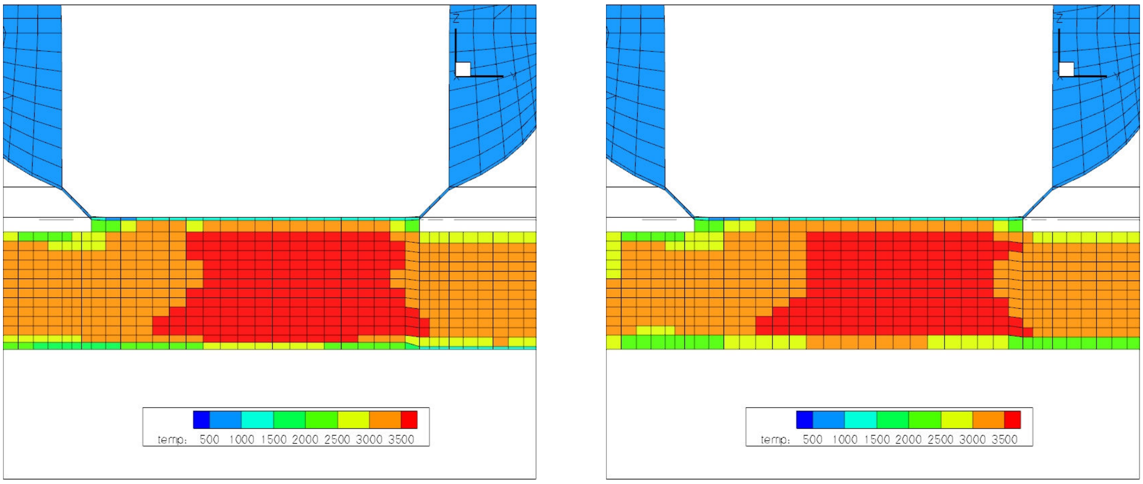


Figure 8.25: Center cell temperature values near TDC compression stroke: before and after snap

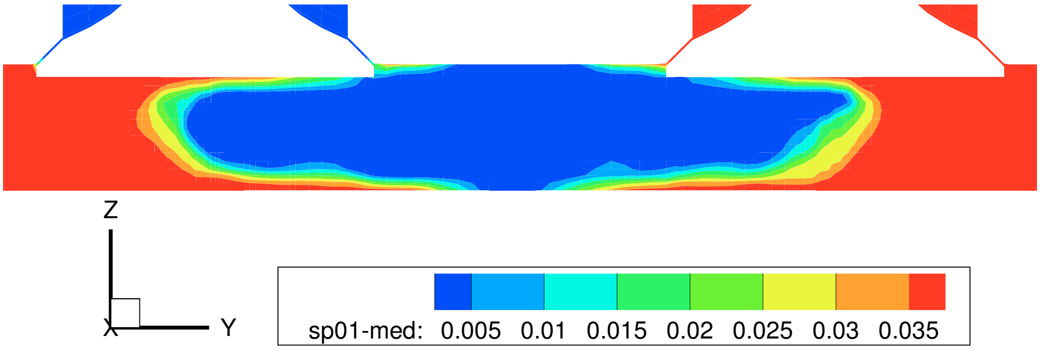


Figure 8.26: Averaged fuel mass fraction 10CA after ignition

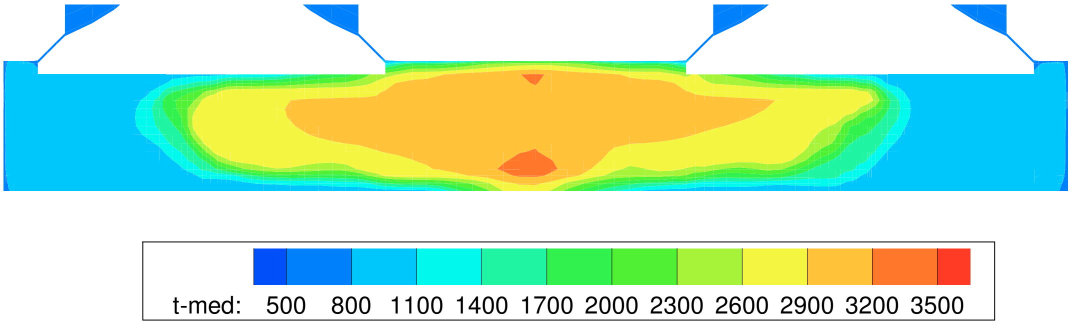


Figure 8.27: Mean temperature 10CA after ignition

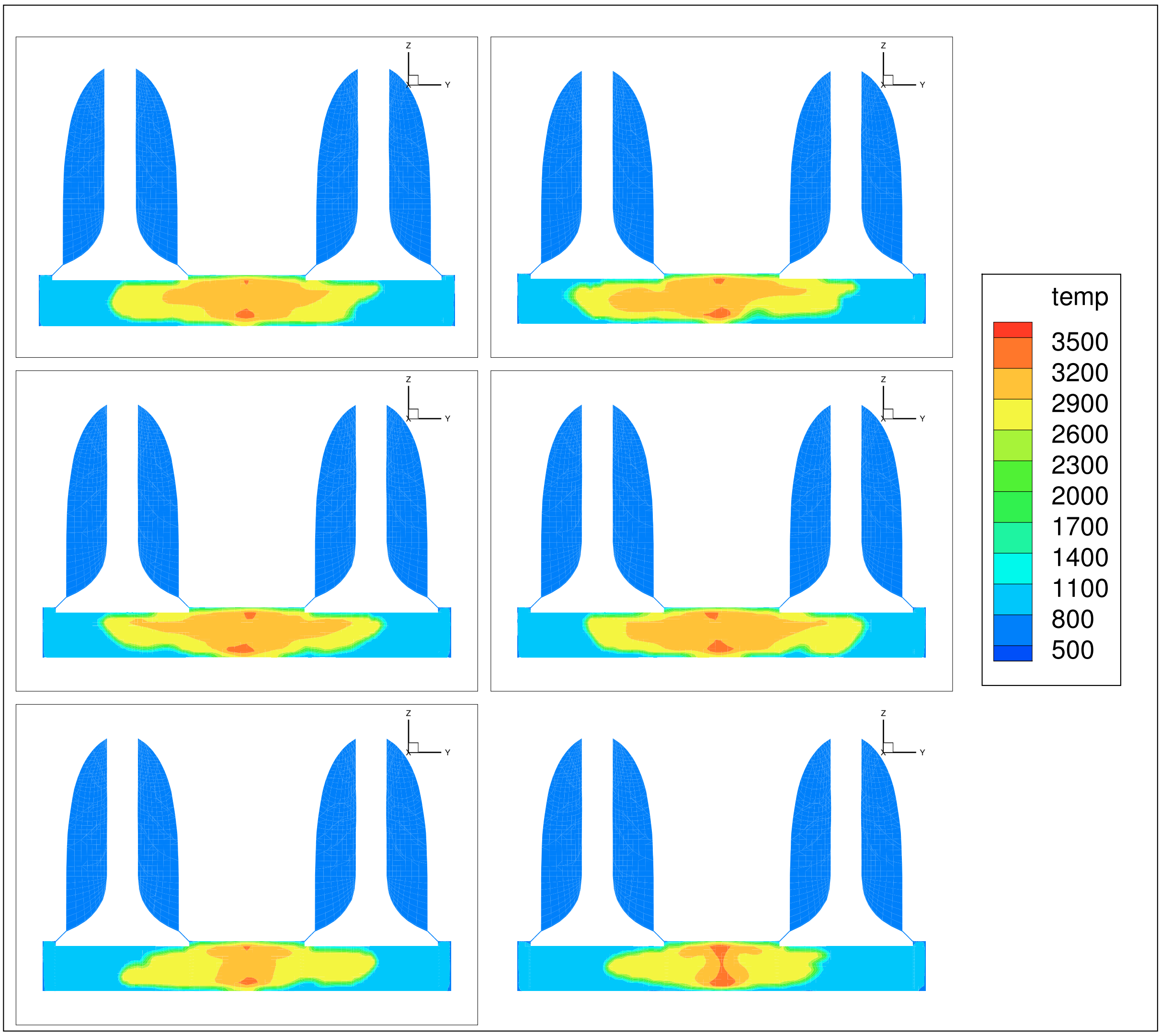


Figure 8.28: Cyclic variation of temperature 10CA after ignition

8.4 Conclusion

This test case was run to validate in detail LES engine simulation with KIVA-4MPI and in code implemented models. Due to few computational capacity and big amount of time spent to find a right grid configuration method suitable for the code, the collected data set is not sufficient to have a robust validation. A complete validation needs a data set at least hundred cycles.

The simulation was run using Smagorinsky model because, Germano model validation still have to be run. The fired case first results demonstrate the possibility to run this kind of simulation with implemented ATF model. However data set is not sufficient large to understand physics inside the process. Besides boundary condition were not apply correctly because fired experimental data were not available.

Although validation is incomplete good results were presented and those can be improved just with a bigger data set.

Chapter 9

RSM engine

The second presented engine configuration is the experimental setup built by RSM (Reaktive Strömungen und Messtechnik) institute at TU-Darmstadt.
This configuration was chosen because of the intensive communication with the experimental team and the comphrensive date set collected in motored and fired operations.

9.1 Configuration

Setup for this kind of simulation was cumbersome because of complexity of the geometry and the elevate values of maximum valve lift. The first issue was produce a quality grid configuration for that case solved with a higher amount of cells than in other cases already studied with the code.
For this mesh, a grid generation concept was utilized to allow a moving mesh during compression, expansion and valve lift. Further details about grid generation and moving mesh could be found in Chapter 7.
The scope of this simulation was ensure successful implementation of the new moving algorithm as well as code validation using experiment results under motored operation. At last step will be fired operation validation, with the help of experimental results.
Some basic engine data are resumed in Table 9.1.

Table 9.1: Basic engine data

Cylinder bore	$86mm$
Piston stroke	$86mm$
Combustion system	4-valve pent roof GDI
Swept volume	$499.6cm^3$
Engine speed	$800rpm$
Inlet valve open	686 ATDC
Inlet valve close	234 ATDC
Exhaust valve open	466 ATDC
Exhaust valve close	14 ATDC

RSM configuration is a single cylinder four canted valves engine with optical access through liner and piston. Injection system is GDI but is not employed in this work. Ignition takes place between spark plug electrodes within the central cylinderaxis.

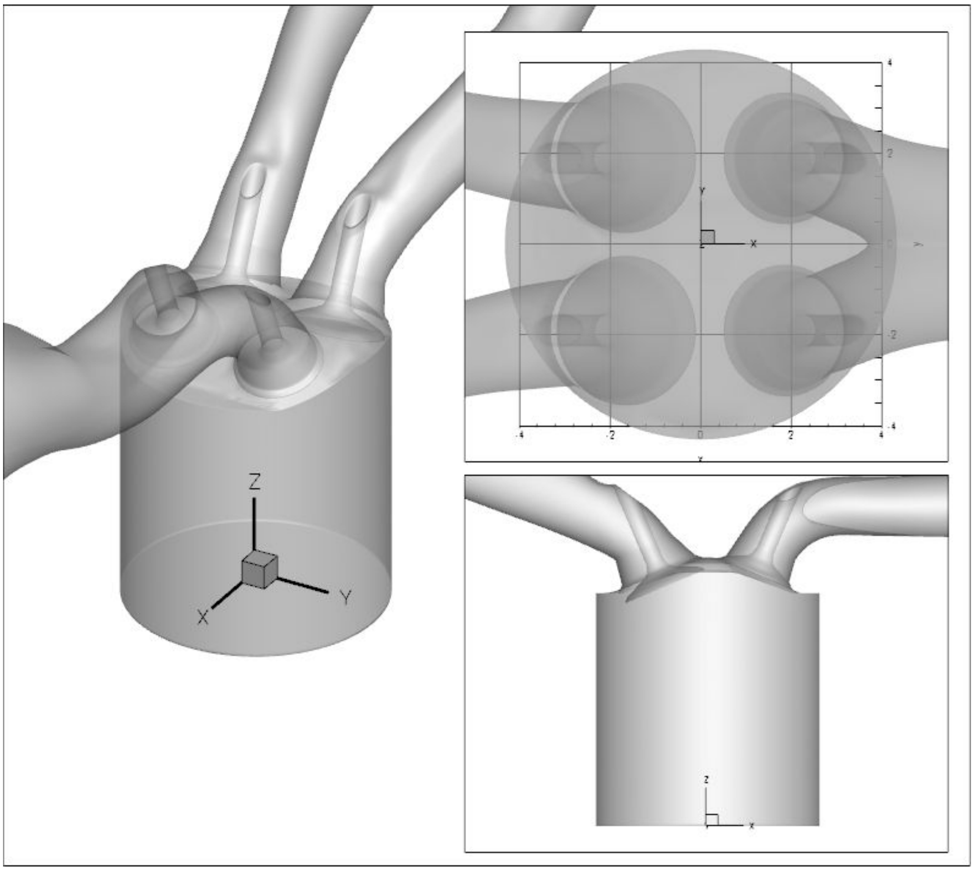


Figure 9.1: RSM engine geometry

Computational mesh is composed by ca. 1.4M cells and ca. 1M of them are inside cylinder volume. Averaged filter size in-cylinder is $0.8mm$.Final computational domain is shown in Figure 9.1.
Boundary conditions are the same collected during experiments, value and distribution diagram is plotted in Figure 9.2.
Because of the geometry complexity some changes that make discretization easy were performed, the most important one and maybe the one that brings some discrepancies on results is the modification of cylinder head in correspondence of injector position as shown by Figure 9.3, where blue surface is the original and measured one and computational grid is the green surface. Omission of this part cause a volume reduction in cylinder but it hopefully won't influence a lot in-cylinder flow. Of course it will be better to include this part but the grid was created wasn't possible to have a good cells quality including this volume.

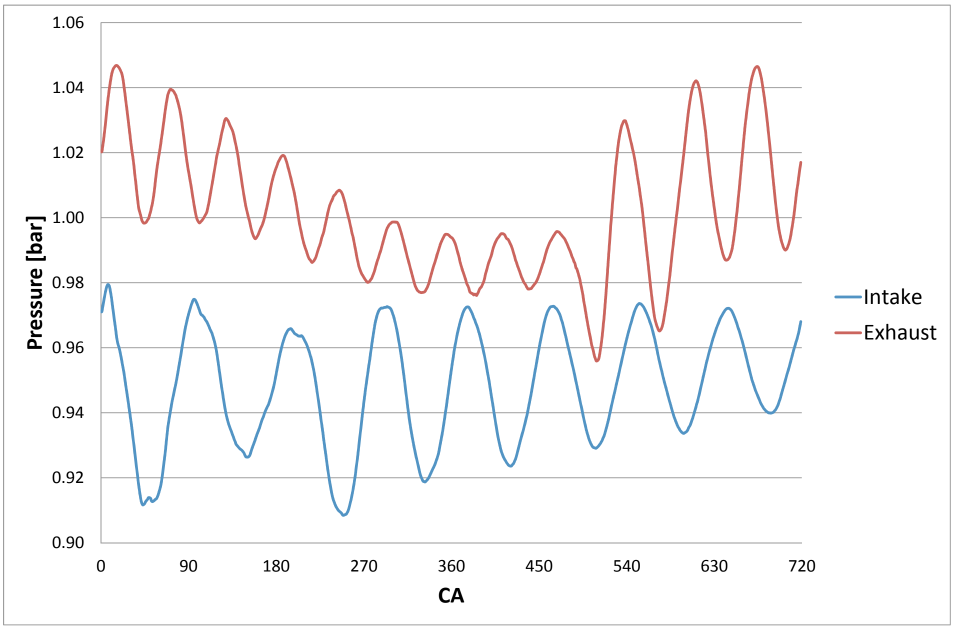


Figure 9.2: RSM engine open boundary condition

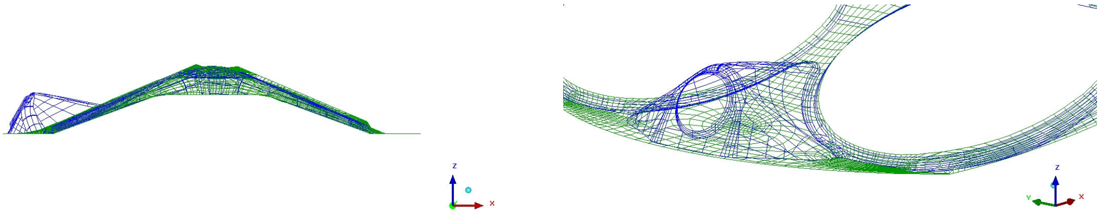


Figure 9.3: Discrepancies between discretized domain and real geometry

9.2 Results

In this section results of the calculated cycles will be reported and compared with experimental data collected by Baum and Peterson [3], [27] and [4].
First compared parameter is the in-cylinder mean pressure, this is basic quantity to verify the correctness of the geometry and boundary conditions. Plot of Figure 9.4 shows a simulated pressure higher than measured one, this is caused by volume reduction described above. A 0-D calculation could demonstrate how in this case 4 or $5cm^3$ volume change maximum pressure of $1.5bar$. In this simulation it was chosen to keep original piston position, so volume reduction is influencing 0-D validation.

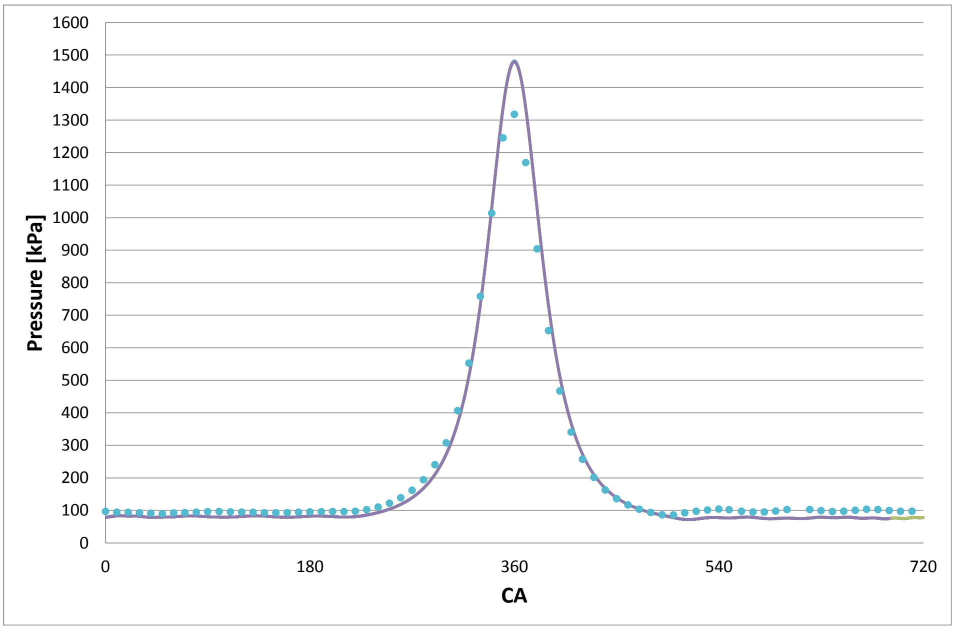


Figure 9.4: RSM engine in-cylinder pressure

Furthermore two dimensional velocity measurements were collected and, as already done for GM engine, simulation compared with experiment.
Data comparison perform along 7 vertical and 4 horizontal lines. This line positions are showed in Figure 9.5 blue lines. Section's coordinate are $x = 0,-10,-20,-30,10,20,30$ for vertical lines and z with $0,-10,-20,-30,-40$ for horizontal, where $z = 0$ is the first section on the top. Mean velocity profile for the simulation experiment comparison along the lines will be performed. The big the number of plots impose a different format in the pages, while the horizontal sections will be disposed in a unique column sorted by z coordinate to result easy to read, vertical section plots will be disposed in three lines, the first one with plots where $x < 0$, the second one with $x = 0$ plot and the last one with $x > 0$ plots. For vertical sections two velocity components will be plotted side by side and for horizontal section the components are plotted in different figures consecutively.
Results are compared on 3 CAD position: one for intake stroke on 100ATDCE and two for compression stroke on 270ATDCE and 315ATDCE.
Afterward contour plots of velocity magnitude for every position will be showed and

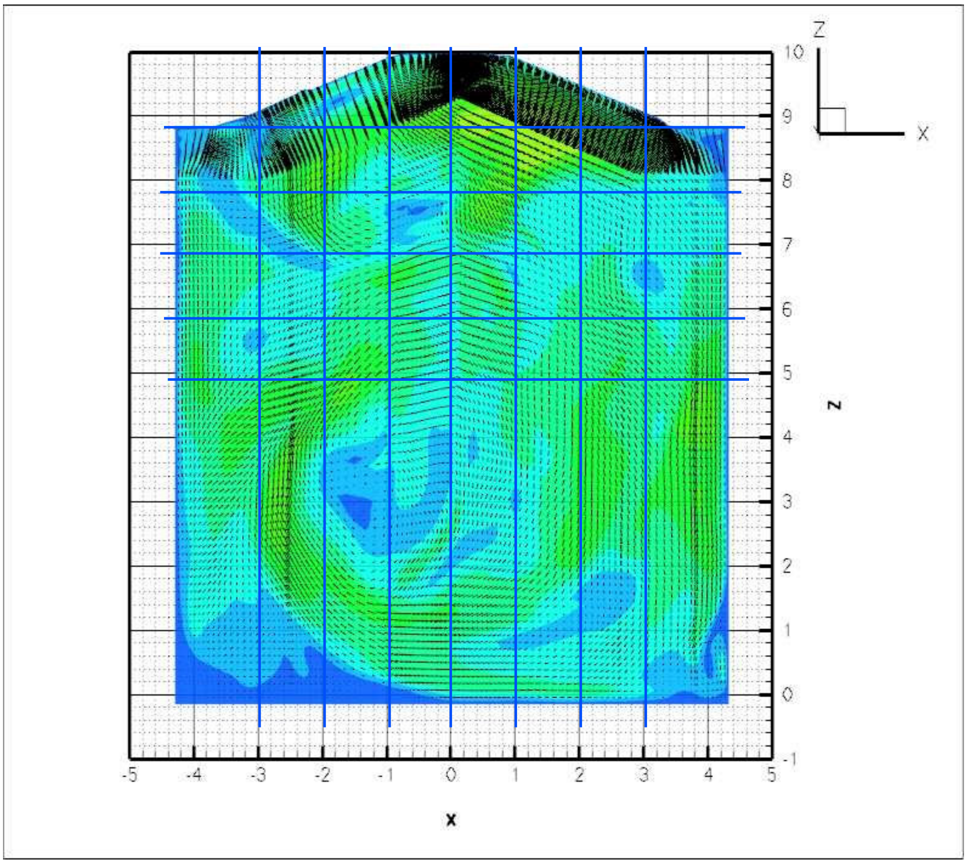


Figure 9.5: RSM engine compared sections

compared with experiments to have a graphic idea of flow direction and distribution. This second comparison is much subjective as the first one because gives less informations, here information about flow large-scale structures as tumble center could be obtained in case simulated cycles are enough to have good quality statistic.
Important notice for the reader is that on the contour plots z-axis between simulation and experiment have a different zero position, for simulation zero is in correspondence of BDC and for experiment is on TDC. The difference between simulation and experiments is ca $88mm$. This difference could be noticed in contour plot but in velocity plots the reference is the same. Besides contour plots for simulation are plotted in cm when simulation in mm.

100ATDCE

As done for previous studied engine, also in this case first analyzed position is 100ATDCE in the middle of intake stroke.

Figure 9.6 and Figure 9.7 show two velocity components in vertical sections, here there is a quite good agreement with experimental results with the exception of the plots $x > 0$ where a slightly difference appears. This could be caused by few calculated cycles and elevated fluctuations or by the geometry changes. In particular major differences present much bigger velocity gradients than experiments, of course biggest factor could influence is the smaller section in this area. Flow in this zone has the same mass but smaller volume, consequence is bigger velocity like in a reduced section channel. Nevertheless other factors of influence are present, so can't be sure geometry modification causes 100% of velocity differences, few simulated cycles is another big candidate for velocities gradients.

Plots in Figure 9.8 have also a quite good agreement with experiment. Exceptions here are plots on $y = -20mm$ and $y = -30mm$ where it seems the gradients are different. Here seems larger flow structure are not captured into simulation. Fluctuations in this position are really high because of tumble formation so more cycle are needed to converge.

From contour plot in Figure 9.9 is possible to see the formation of tumble motion with center located circa in the positions where plots have more problems to fit. This confirm the supposition advanced with plot about zone affected by high fluctuations. Other difference could be noticed is the lower velocity magnitude below tumble center. In this zone there is a strong influence of near wall simulation and grid dimension.

Comparing (Figure 9.9) simulation and experiment is clear that the convergence in the simulation is not reached but recirculation zone could already be seen. Velocity near valves seems to be too high and this could be a grid resolution issue not just near valves, where the grid is much finer than needed, but also below where grid become coarser. Grid coarsening could also be a problem for tumble center position calculation. In this case could be suggested to run more cycle to understand the convergence of the mean velocity flow field, followed by a finer grid and compared with the grid computed.

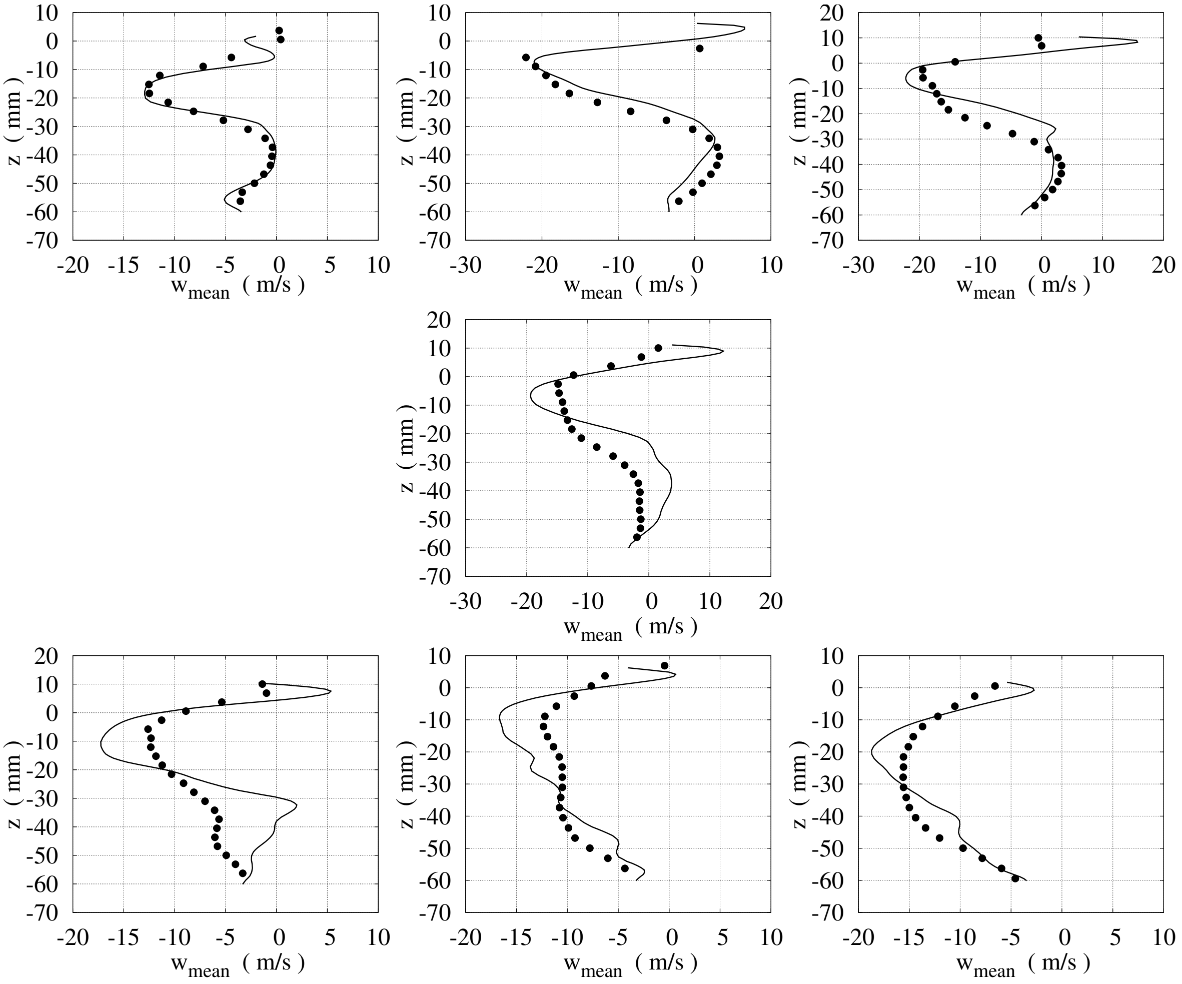


Figure 9.6: Component w of the velocity in vertical sections

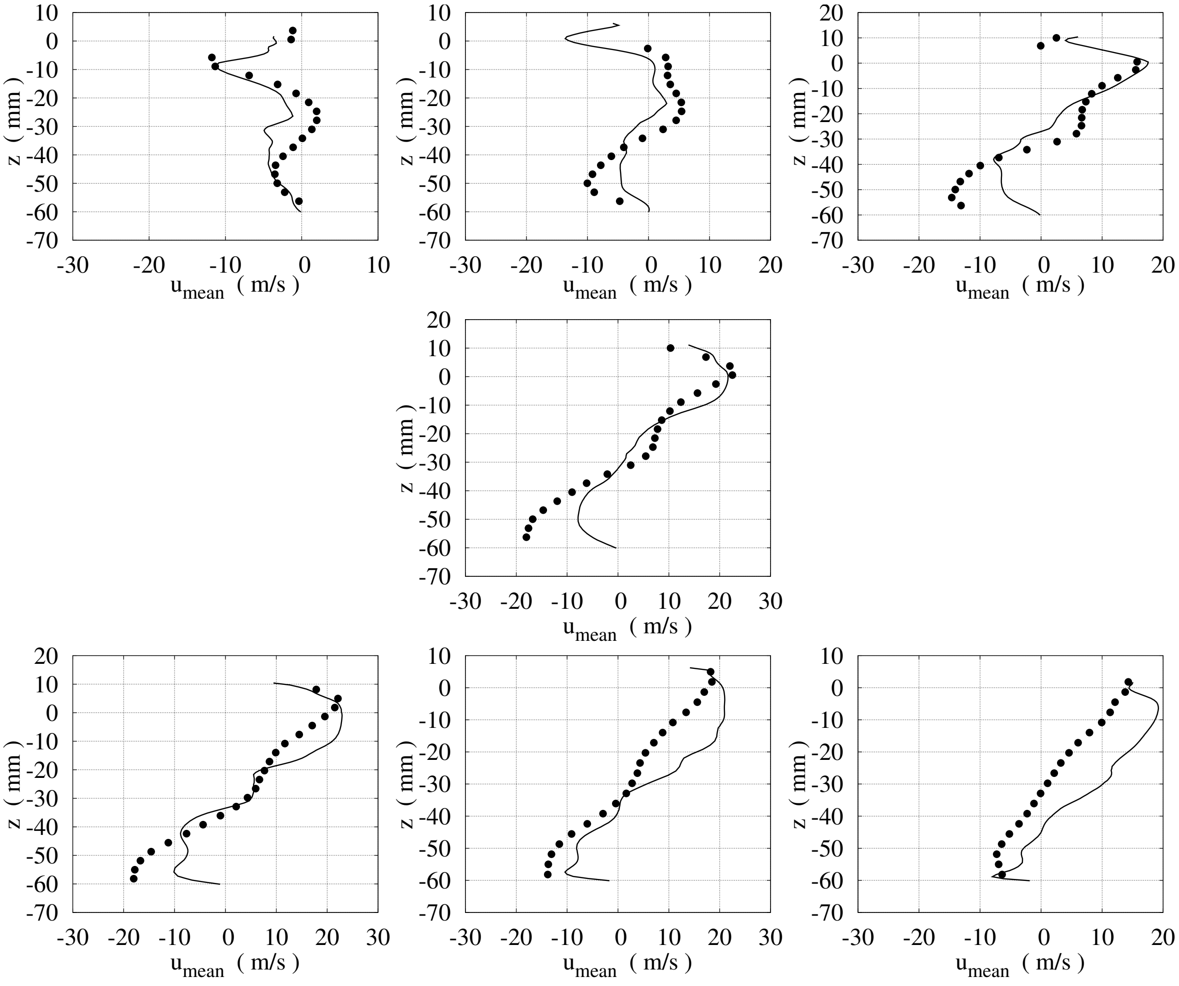


Figure 9.7: Component u of the velocity in vertical sections

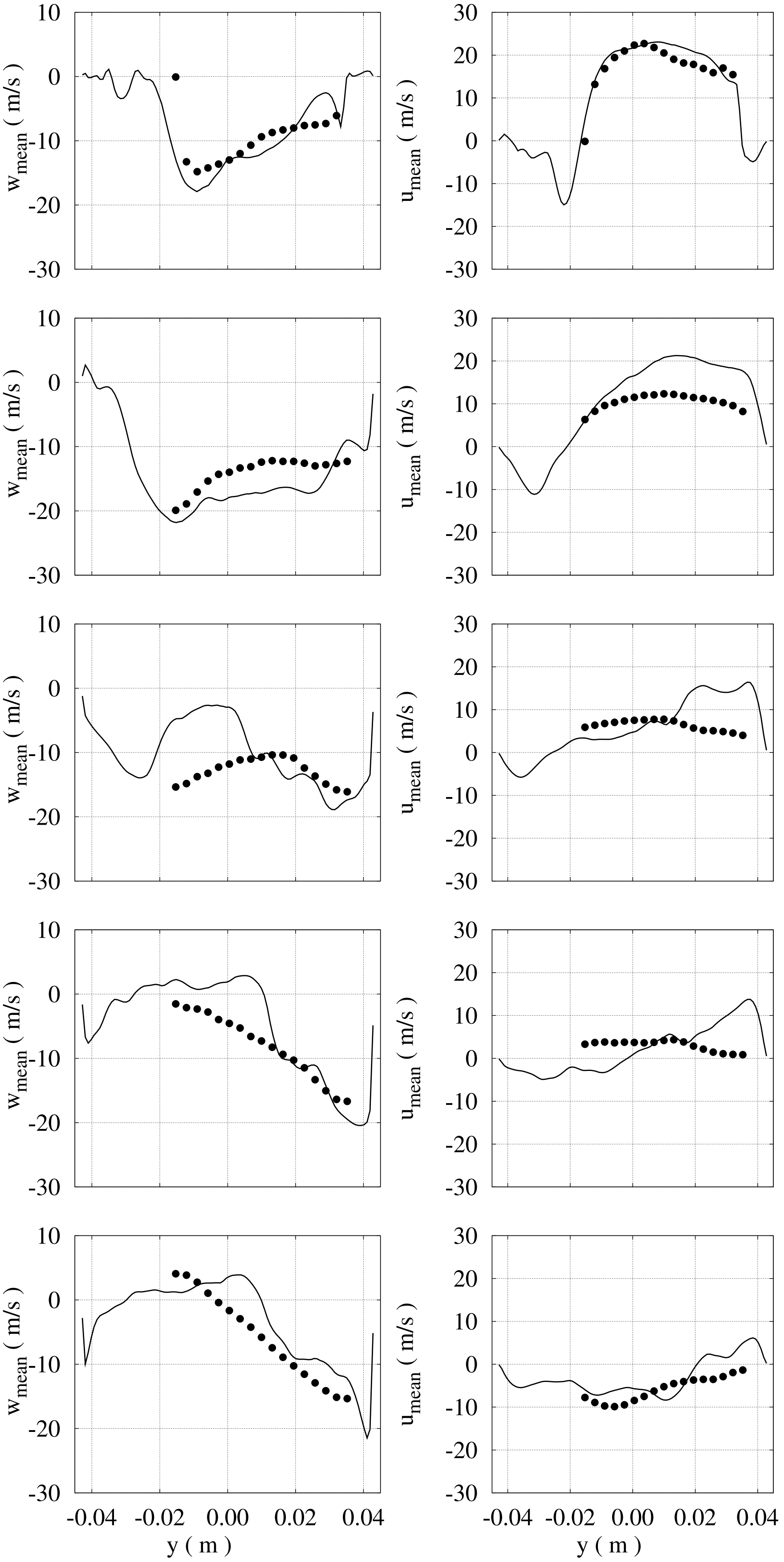


Figure 9.8: Two components of the velocity in horizontal sections

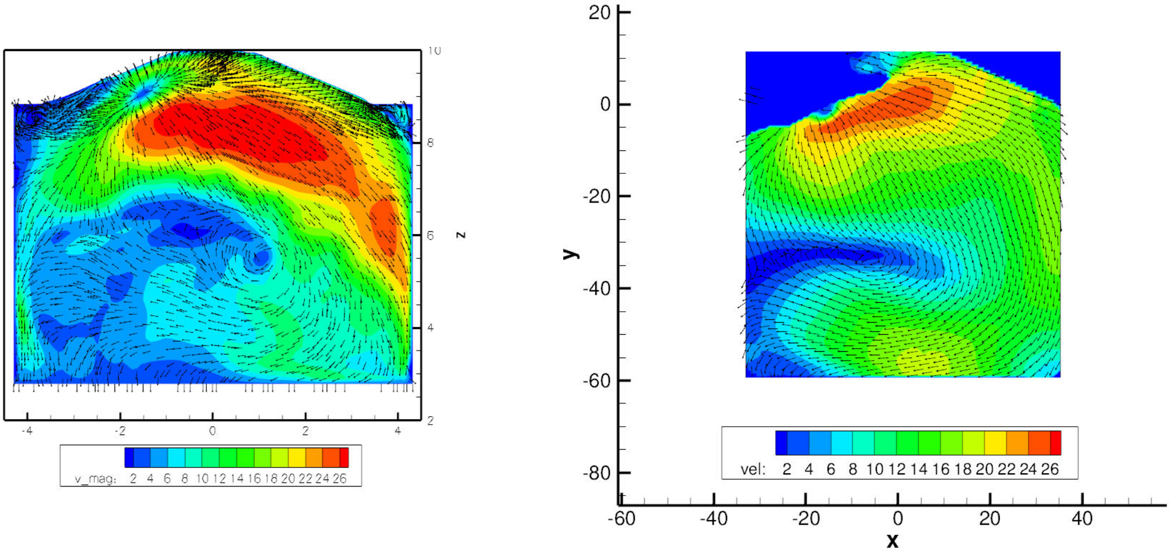


Figure 9.9: Contour plot and vectors of average velocity: simulation (left) vs. experiment (right)

270ATDCE

This position is on the first half of compression phase, intake valves are closed since 40CA and the piston is half way to the TDC. Velocities in this position are slower than at $100ATDCE$, which makes fluctuations relatively higher than previous and plots less precise.
Plot showed by Figure 9.10 and Figure 9.11 are not fitting with experiment in particular in $x > 0mm$ zone where tumble center is located. This is moved on the left side and the w component of velocity in plot $x = 20mm$ have the reverse direction than experimental one. However u component fit quite good with experiment in all plots but last one on $x = 30mm$ where may the tumble center position influences the simulation. Figure 9.10 is the clear example how the geometry could influence the flow on slower motion. Where geometry does not change $x < 0$ gradient are similar but velocities much bigger. On the other hand where geometry was modified, $x > 0$, the gradients are much different from experiment.
All those differences cannot just be velocity variance due to few calculated cycle but a strong change in the flow motion due to the geometry variation.
Other component fit much better because is not influenced from this factor.
Figure 9.12 also shows a clear motion of tumble center to the center of the geometry that bring a strong gradient of w component near $x = 0$ Otherwise velocities fit quite good with experiments.
From Figure 9.13 while comparison with experiment, one could see that tumble center is moved to the center of $5 - 10mm$ and $2 - 3mm$ to the bottom besides one could notice a smaller dimension of tumble center on experiments. This phenomena is caused just from high fluctuation. Other interesting aspect in this Figure is the different velocity magnitude in the middle-left hand side of cylinder; while experiments show a quite slower motion, simulation has $2ms^-1$ more in the magnitude. As said for previous analyzed position some issues could be solved using a slightly finer grid, especially on the lower zone of cylinder.

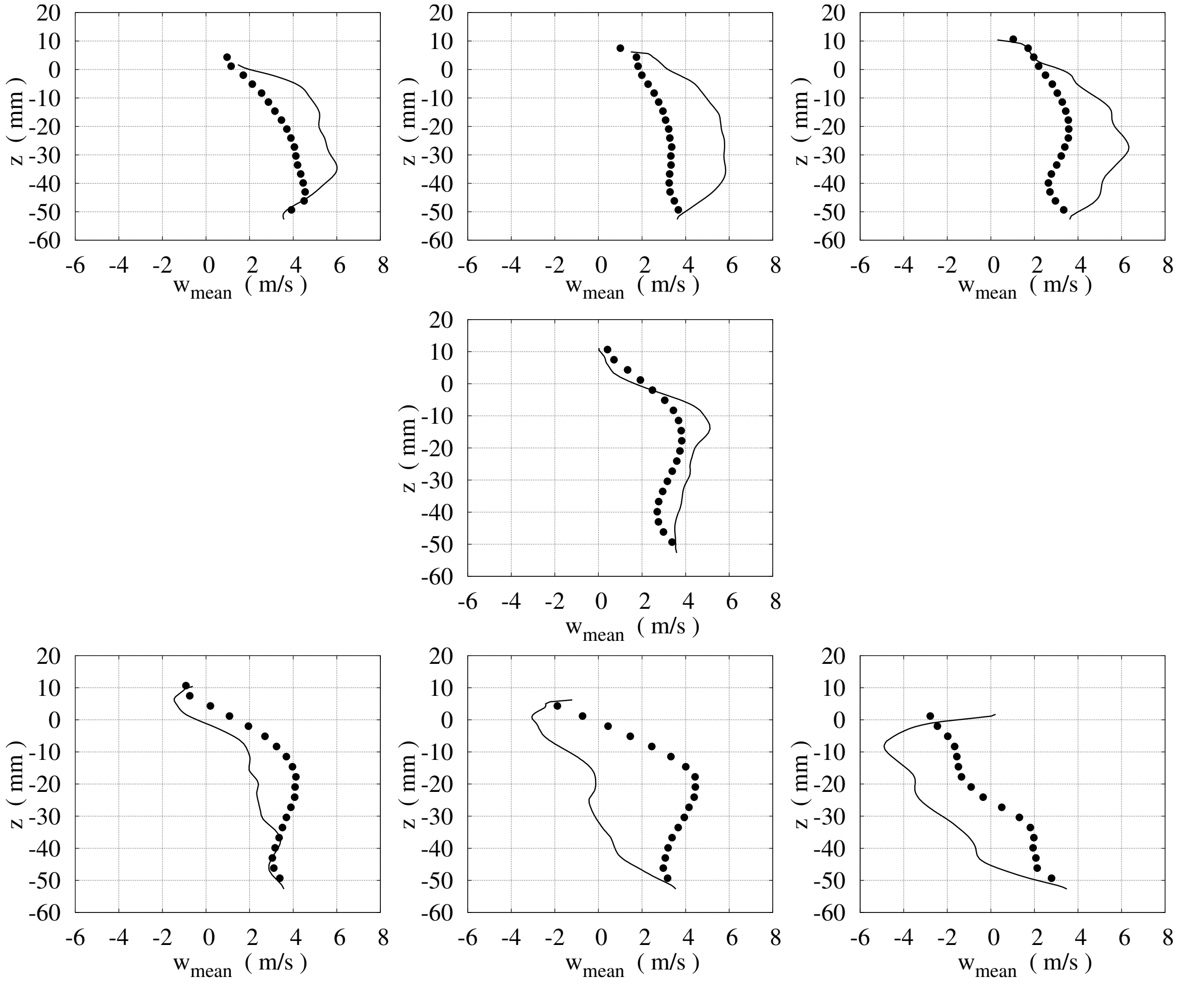


Figure 9.10: Component w of the velocity in vertical sections

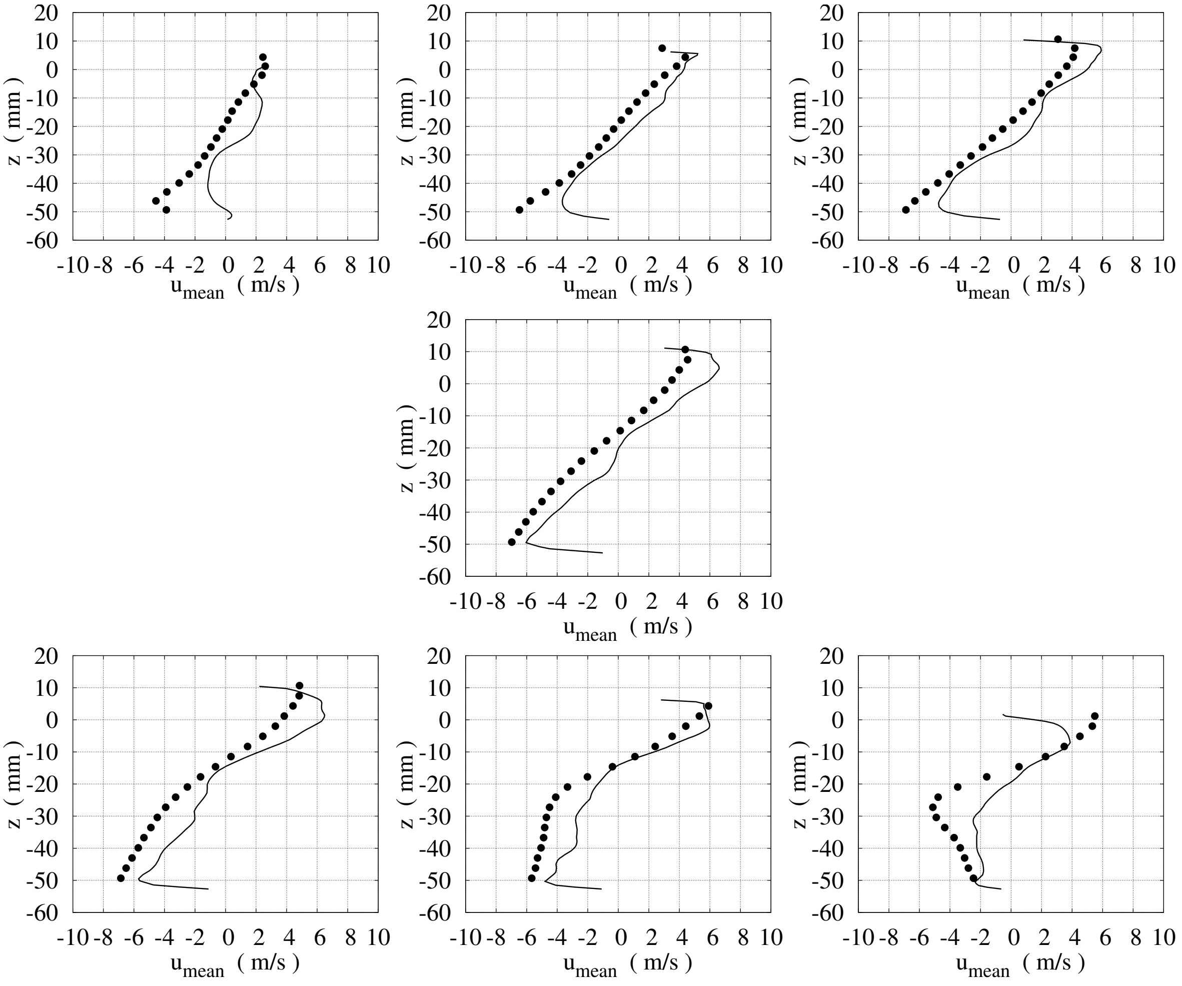


Figure 9.11: Component u of the velocity in vertical sections

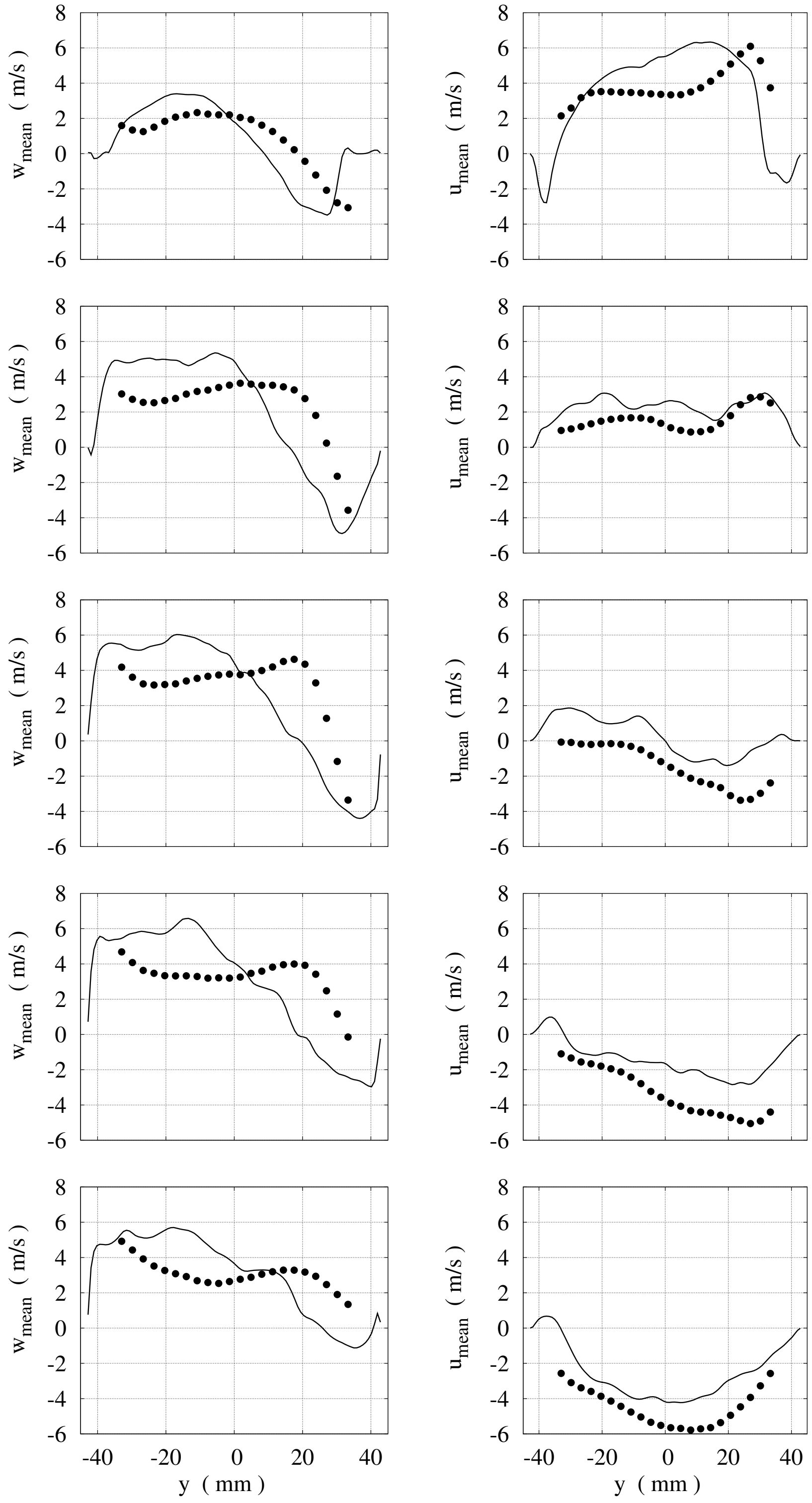


Figure 9.12: Two components of the velocity in horizontal sections

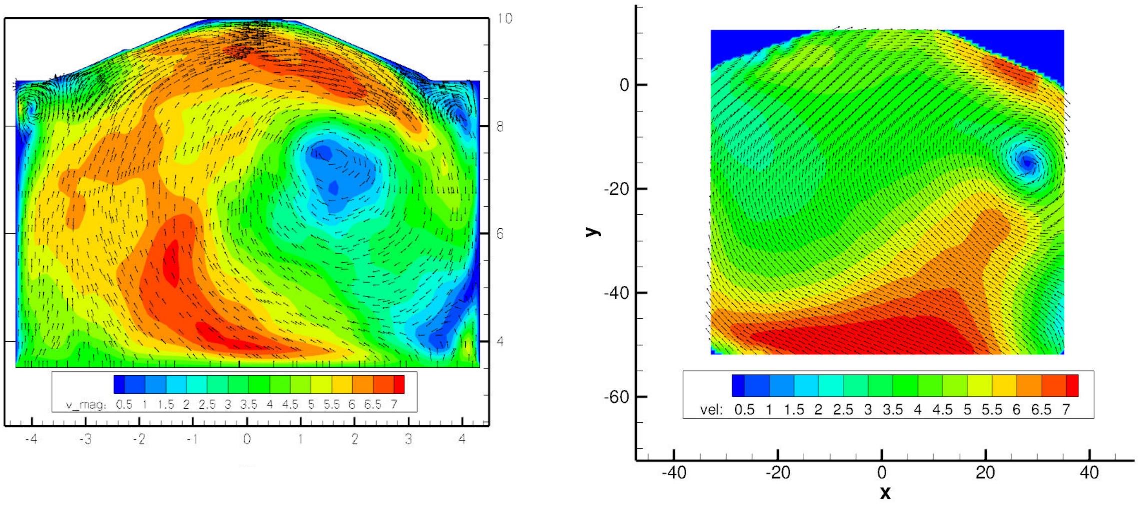


Figure 9.13: Contour plot and vectors of average velocity: simulation (left) vs. experiment (right)

315ATDCE

Last analyzed position is also during compression stroke but at the end, 45CA before TDC. This position is really important because in fired operation ignition will start near this position (16CA BTDC) so flow configuration will be important for flame propagation.

In this position, after the whole compression stroke tumble is dissipated and quasi disappeared, velocities are slightly smaller than before due to dissipation process. Figure 9.14 shows a quite good match between experiments and simulation also near critical sections studied in previous position. Smaller velocities exist for the simulation and can be dive to the dissipation as discussed in Chapter 6. It should also be mentioned that the grid is the coarsest at this location and may lead to the smaller velocities.

Figure 9.15 confirm the above remarks and velocities fit quite well with experiments but there are always slightly smaller.

Figure 9.16 is reduced, due to reduced cylinder dimension, so just two sections are reported. Here results from previous sections are repeated, here another clear example about higher dissipation is in the first u plot, where experiments and simulation have same distribution but simulation gradient are much smaller and computation curve results flatter. Figure 9.17 shows, besides strong velocities reduction already threated before, a quite good positioning of the tumble center, located circa in the same position of experiment. As previously mentioned flow large structures are important in this position not just in this case but also when combustion will be activated.

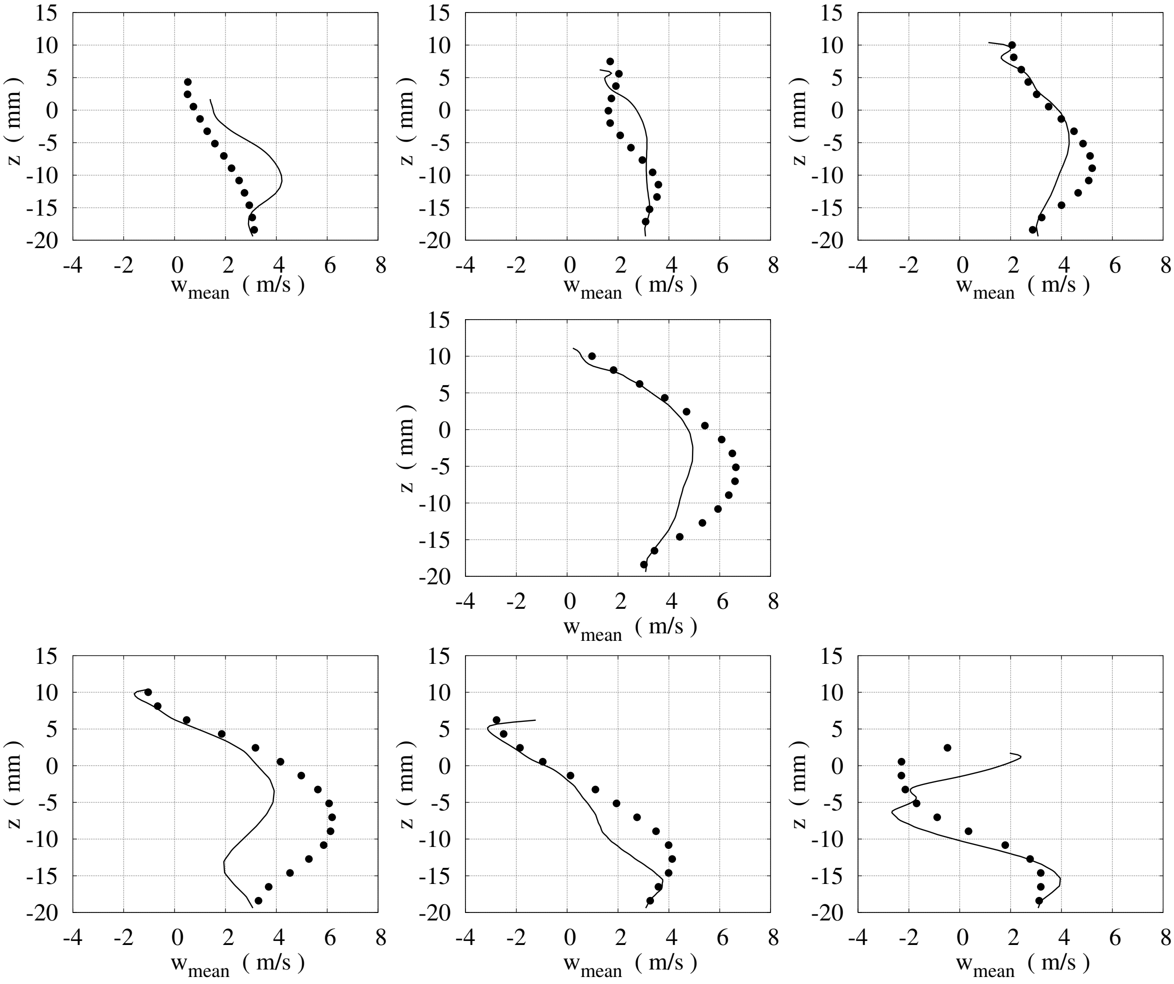


Figure 9.14: Component w of the velocity in vertical sections

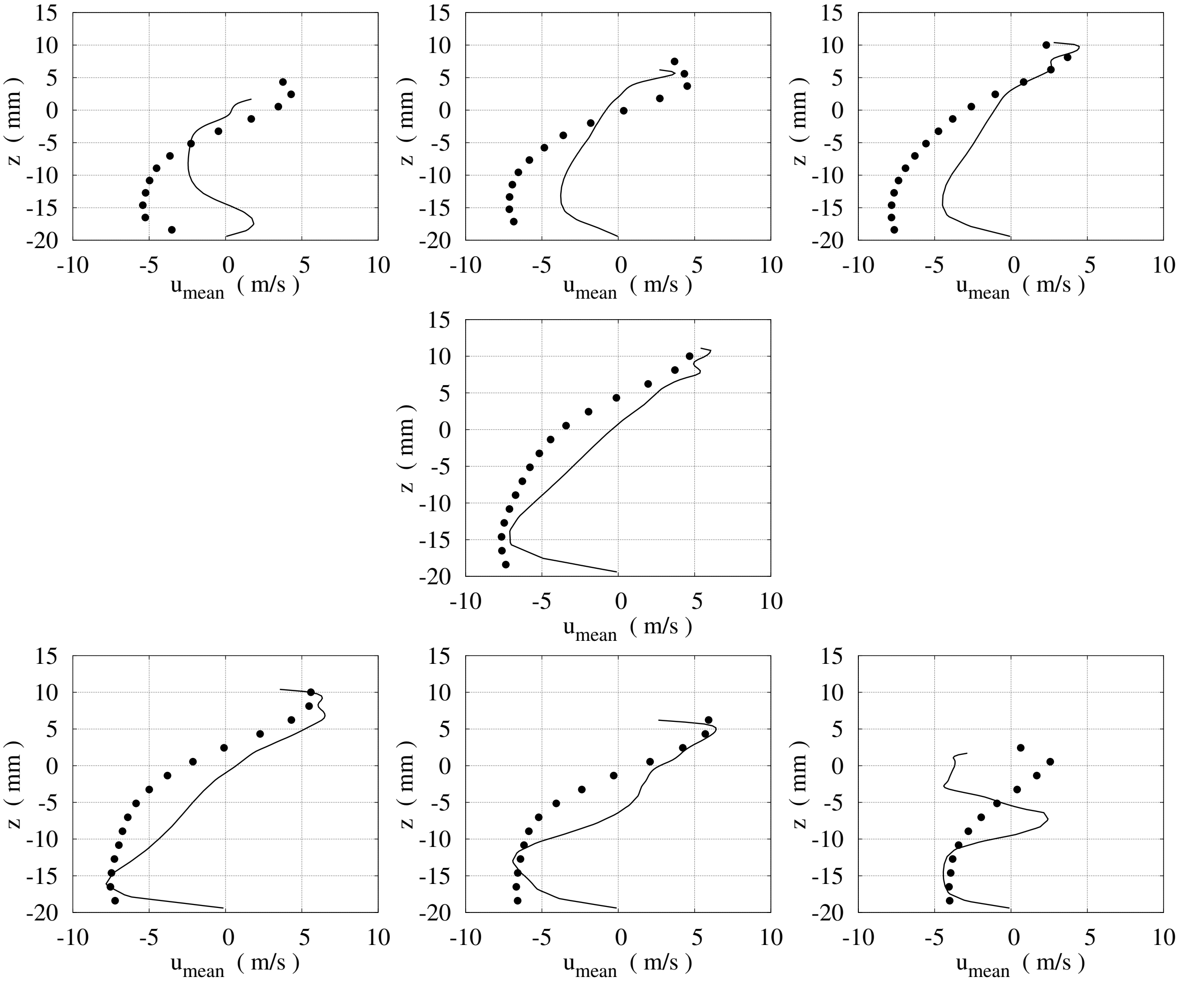


Figure 9.15: Component u of the velocity in vertical sections

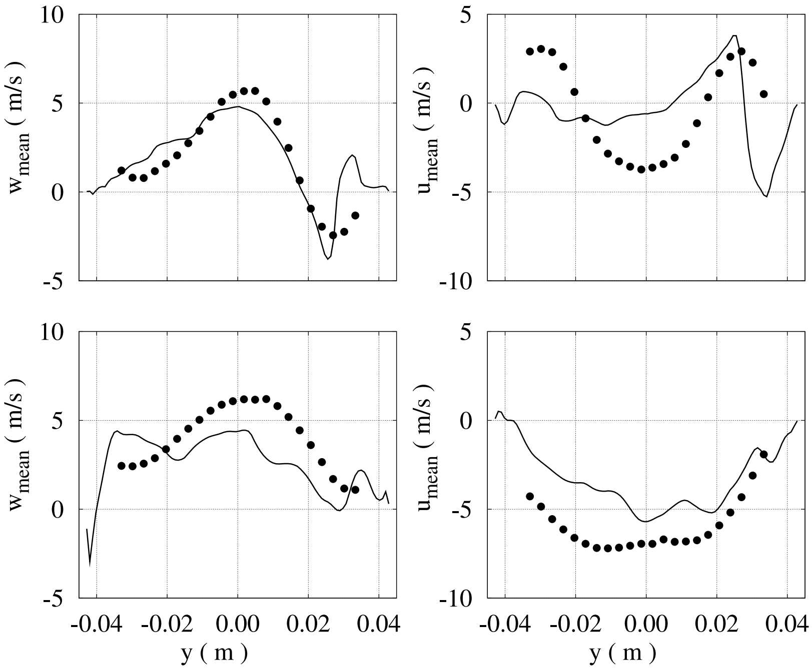


Figure 9.16: Two components of the velocity in horizontal sections

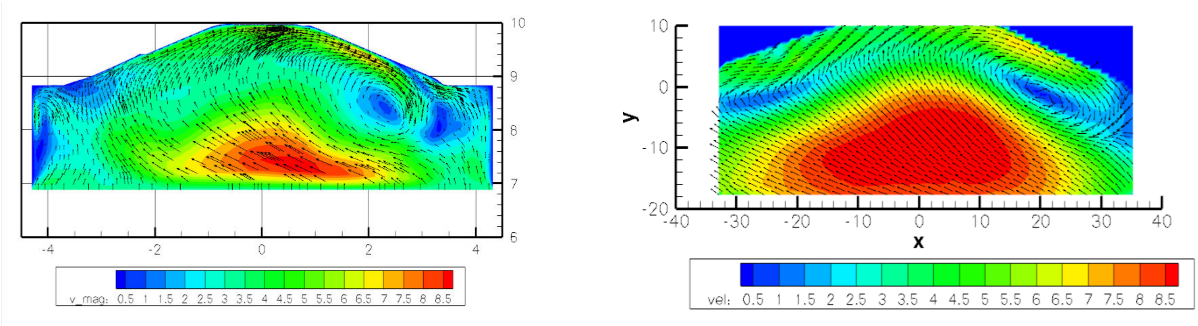


Figure 9.17: Contour plot and vectors of average velocity: simulation (left) vs. experiment (right)

9.3 Fired case

This engine was also studied in fired operation and a large data set of experiments is available. When this work was written some results were available like in-cylinder pressure and flame position for some positions. Unfortunately CFD code development didn't reach the point where fired simulation could be easily done.

Otherwise some fired simulations were run and some considerations can already be done. In this case boundaries informations are available and the correct boundary condition are applied. Pressure distributions on the boundaries are shown in Figure 9.18. Fired case has a different pressure distribution on boundaries than cases presented by **??**. Particular oscillation can be seen on outlet where after exhaust valve opening introduces a high amplitude wave in the channel.

The chosen fuel is iso-octane and its oxidation will be simulated with one step chemistry coupled with other nine reactions to stabilize combustion.

Used reactions are:

$$\begin{aligned}
2\,C_8H_{18} + 25\,O_2 &\longrightarrow 16\,CO_2 + 18\,H_2O \\
O_2 + 2\,N_2 &\longrightarrow 2\,N + 2\,NO \\
2\,O_2 + N_2 &\longrightarrow 2\,O + 2\,NO \\
N_2 + 2\,OH &\longrightarrow 2\,H + 2\,NO \\
N_2 &\longleftrightarrow 2\,N \\
O_2 &\longleftrightarrow 2\,O \\
H_2 &\longleftrightarrow 2\,H \\
2\,CO_2 &\longleftrightarrow 2\,CO + O_2 \\
2\,OH &\longrightarrow O_2 + H_2 \\
2\,H_2O &\longrightarrow 2\,H_2 + O_2
\end{aligned}$$

Injection during experiments was done before intake channel, so perfect mixture is entering in simulated domain. Mixture is ignited through spark plug 16CA BTDC. The spark is not included inside discretized domain, but simulation needs a volume where the energy source is introduced. So the volume occupied by spark terminal was measured and defined as ignition volume. In this case simulation start point was decided on -180CA to simulate an exhaust and intake strokes before ignition. This choice was done to verify if half a cycle is sufficient to simulate a first cycle combustion similar to the following combustions.

As noticed with previous fired engine some issues are presenting while exhaust valve is opening. In this phase temperature and pressure jump between in cylinder flow and exhaust port brings high velocities through valves. Time step, due to Crank Nicholson, becomes very small and simulation spends circa $30 - 40\%$ of its total cycle calculation time within few CA when this phenomenon happens.

Due to this issue one complete cycle calculated with 12 cores needs three weeks to be completed. The used of more core was not possible to apply because of CFD code parallelization issues.

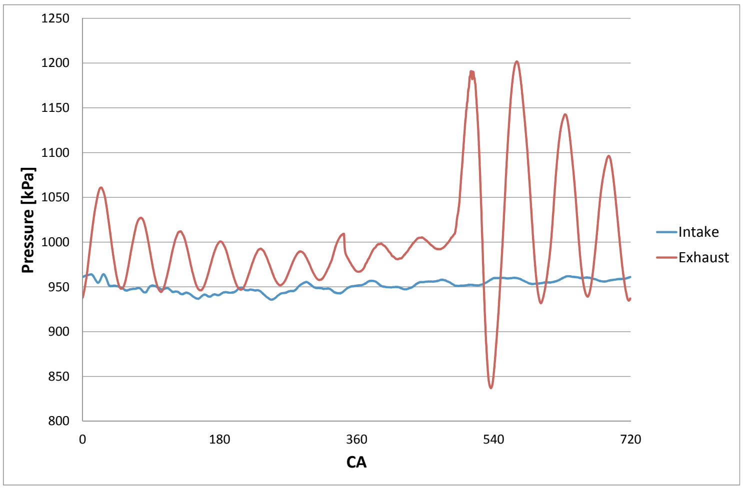


Figure 9.18: Pressure boundary condition for fired operation

Unfortunately just two cycles were calculated at the end of this work, so presented results cannot be considered a robust. However some screen shots of the combustion process will be presented. Those plots give an idea of the flame propagation during this process. Besides the results can suggest necessary improvement needed in the code the increase the simulation quality.

Figure 9.19 and Figure 9.20 show two consecutive calculated cycles. Reference cycle is the second showed in first Figure. First big difference between two cycles is the combustion timing lag. While second cycle completes combustion $20 - 22$CA after ignition, first cycle needs at least 35CA after ignition. This can be explained by the strong difference due to initialization especially for flow structures and temperature.
Other important remark, easy to see in early combustion phase, is about flame propagation near walls, that seems not really good simulated. Near the wall the flame needs very thin cell to be computed, otherwise structures and gradients cannot be properly calculated. The grid could be refined on the wall with code procedure implemented by the author, but in this case this refining was deactivated.
For quantitative results more cycles must be collected and to compare simulation with experiments also a refined grid must be calculated.

9.4 Conclusion

In this Chapter a second case of multicycle engine simulation was presented. Setup for this simulation was much complicated than TCC-GM. In this case grid generation took several month, but the most time consuming procedure was to modify motion algorithm included in original code. This modification is flexible and it can be applied on every engine.

In both simulated cases few cycles were simulated and the data set is not sufficient to give robust results. Otherwise the non-reacting case comparison have a quite good match with experiments. If sufficient data can be collected, computation won't be far from experiments.

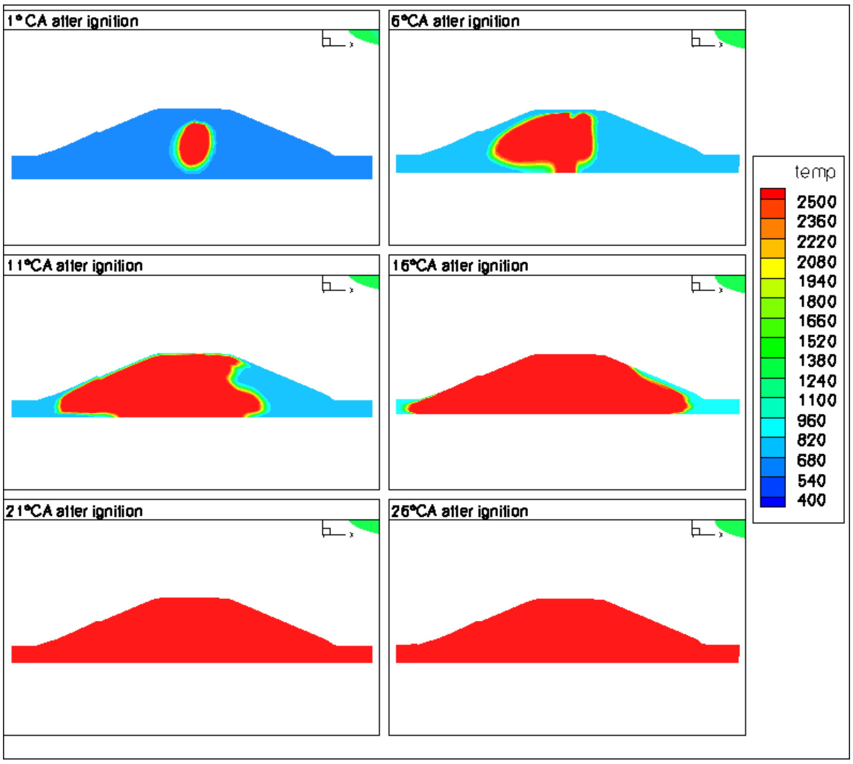


Figure 9.19: Flame propagation during second combustion cycle

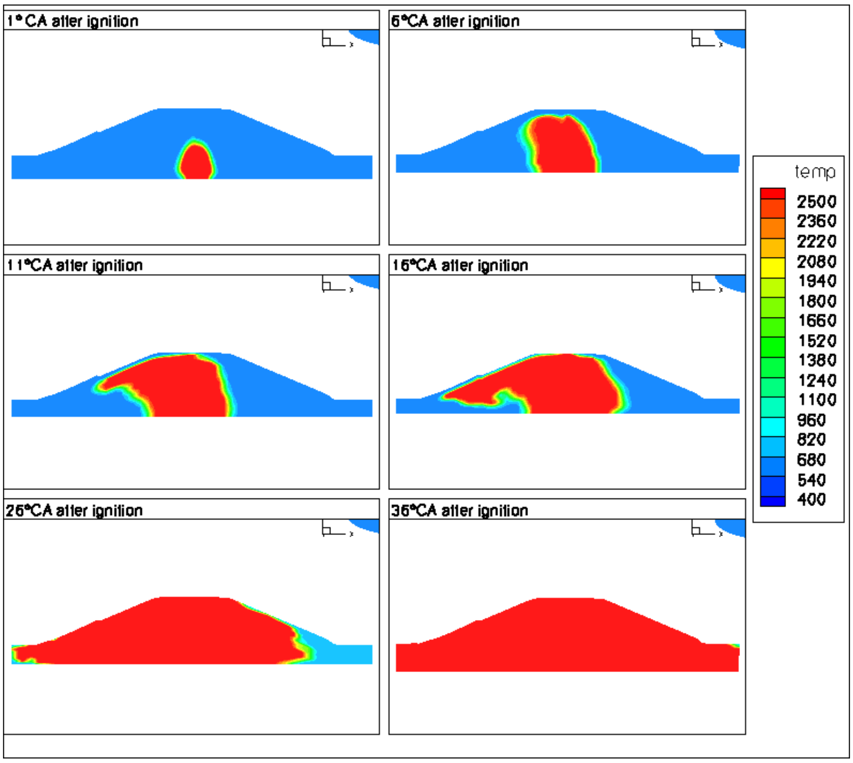


Figure 9.20: Flame propagation during first combustion cycle

Chapter 10

Conclusions

In this work a numerical method for analysis of IC engine in LES context was developed, verified and validated.

This work shows the complexity of this kind of simulations, which peculiarity is the moving domain necessary to simulate a complete cycle including combustion.

The most important and time consuming part of the work was find and apply a new method of grid generation. The resulting procedure allows to create grid for every kind of IC engine with single cylinder. However sometimes the method results difficult and cumbersome. During the work, two geometries were meshed using the developed method. Both grids were simulated and results are reported in the Chapter 8 and 9.

Although grid generation method is very efficient and can handle with every kind of engine, sometimes there are limitations. This work introduces and implements a new concept of grid movement, which can manage the grid during domain motion keeping a constant quality and dimension of the grid. Thanks to this implementation, it was possible to simulate the engine presented in Chapter 9 and which results are satisfactory.

Besides, flow and combustion simulation were improved by new implemented models. While combustion simulation quality without modeling was not very good and the results of those simulation had lack of informations, implementing new model flame can be easily traced and its propagation is more realistic.

Simulation presented in Chapter 6 shows that implemented models give a good outcome. In this Chapter comparison between simulation and experiments is presented and the results fit well. The simulation was done to validate the CFD code with new implemented models.

The simulation presented in Chapter 8 was the longest in terms of number of simulated cycles. Thanks to this simulation, which data set reached 40 cycles, was possible to understand that the CFD code can be handled with the new simulation method, giving results comparable with reality (experiments). Besides the computational request of this method is not high, if compared with other commercial software for the same amount of cells and quality of results. Besides, an actual issue of the code is the limited number of cores available for simulation. This limit increases the time needed to simulate a case. In a world where is important to give accurate results within the shortest time, software parallelization is a must. One of the most important issues for the CFD code at the end of this work is the parallelization strategy.

In the simulation presented in Chapter 9, parallelization issue weight more on results quality. This simulation was much longer than TCC-GM but data set is smaller. A simulated cycle with 12 cores needed one week to be computed and there was no possibilities to reduce the time increasing the parallelization. A cycle calculated including combustion took more than three weeks to be concluded. So long computational time was not expected at the beginning of the simulation and due to grid generation and motion improvement the simulation started few month before the end of this work. For this reason the collected statistics are not robust and it is not possible to give a complete view of the CFD code potentiality with this simulation, but, inside the Chapter, physics seems have a very good match.

The fired cases had the same issue concerning parallelization and those cases were not simulated enough time to have a statistical data set. The combustion simulation was one of the initial objectives of this work and the main reason why the project was promoted. However during the work many problems arose and delayed the begin of a quality simulation including combustion. However those cases are not validated with a robust comparison.

This work concludes with some results but with a lot of point where other projects can start. First issue that must be necessarily solved is the parallelization. Second important issue is about the grid generation, although the new method allows a big range of possible cases which could be simulated, results complicated and time consuming. Therefore, further studies on the method needs to be done. An other open point is the validation of fired cases, which are already started and need more time to produce sufficient larger data set. When parallelization and grid generation issues will be solved, further grids could be calculated and a cell dimension study could be presented. Last suggested issue is regarding the CFD code, which is still complex and not user friendly. Continuing with the improvement process started during this work, in few years this software could become easier to learn and to use, but to do this lot of work still needs to be done.

Bibliography

[1] Amsden, A. A.: *KIVA-3V: A Block-Streuctured KIVA Program for Engines with Vertical Valves or CAnted Valves.* Los Alamos national laboratories, New Mexico, 1997.

[2] Amsden, A. A., P. J. O'Rourke and T. D. Butler: *KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays.* Los Alamos national laboratories, New Mexico, 1989.

[3] Baum, E., B. Peterson, B. Boehm and A. Dreizler: *On the validation of LES applied to internal combustion flow: Part 1: Comprehensive experimental database. In review.* Flow, Turbulence and Combustion, 2013.

[4] Baum, E., B. Peterson, C. Surmann, B. Boehm and A. Dreizler: *High-speed PIV and LIF imaging of temperature stratification in an internal combustion engine.* Proceedings of the combustion institute, 2012.

[5] Borger, M., D. Veynante, H. Boughanem and A. Trouve: *Direct numerical simulation analysis of flame surface density concept for Large Eddy Simulation of turbulent premixed combustion.* Proc. Combust. Inst., (27):917–925, 1998.

[6] Charlette, F., C. Meneveau and D. Veynante: *A Power-Law Flame Wrinkling Model for LES of PremixedTurbulent Combustion. Part I: Non-Dynamic Formulation and Initial Tests.* Combustion and Flame, 2002.

[7] Cocchi, A.: *Elementi di Termofisica generale e applicata (italian language).* Progetto Leonardo, Bologna, 1974.

[8] Colin, O., F. Ducros, D. Veynate and T. Poinsot: *A thickened flame model for large eddy simulations of turbulent premixed combustion.* Physics of Fluids, 12(17), 2000.

[9] Cornetti, G.: *Fondamenti di macchine (italian language).* Signum scuola, Torino, 1997.

[10] D., Butler T. and O'Rourke P. J.: *A numerical method for two dimensional unsteady reacting flows.* 16th Symp. (int) on combustion, Combustion institute, pages 1503–1515, 1977.

[11] Durand, L. and W. Polifke: *Implementation of the Thickened Flame Model for Large Eddy Simulation of Turbulent Premixed Combustion in a Commercial Solver.* 2007.

[12] Ferrari, G.: *Motori a combustione interna (italian language).* il Capitello, Torino, 1992.

[13] Germano, M., U. Piomelli, P. Moin and W. H. Cabot: *A dynamic subgrid-scale eddy viscosity model.* Phys. Fluids A, 3(7):1760–1765, 1991.

[14] Giacomazzi, E., V. Battaglia and C. Bruno: *The couopling of turbulence chemistry ina premixed bluff-body flame as studied by LES.* Combustion and flame, (138):320–335, 2004.

[15] Goryntsev, D.: *Grobstruktursimulation von Stoemung und Mischung in einem Verbrennungsmotor.* PhD thesis, Technische Universitaet Darmstadt, Germany, 2007.

[16] Hahn, F.: *Zur Vorhersage technischer Verbrennungssysteme im Hinblick auf fluessige Brennstoffe.* PhD thesis, Technische Universitaet Darmstadt, Germany, 2009.

[17] Heywood, J. B.: *Internal Combustion Engine Fundamentals.* McGrew-Hill, Inc., 1988.

[18] Hirt, C. V., A. A. Amsden and J. L. Cook. Journal of computational physics, 14(227), 1974.

[19] K., Yeung P., Girimaji S. S. and Pope S. B.: *IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion.* combustionand flame, 3-4(79):340–365, 1990.

[20] Kempf, A.: *Large-Eddy Simulation of Non-Premixed Turbulent Flames.* PhD thesis, Technische Universitï¿$\frac{1}{2}$t, Darmstadt, Germany, 2003.

[21] Laydler, K. J.: *Chemical Kinatics.* Benjamin-Cummings, 1997.

[22] Legier, J. P., T. Poinsot and D. Veynante: *Dynamically thickened flame LES model for premixed and non-premixed turbulent combustion.* Center for Turbulence Research Proceedings of the Summer Program 2000, 2000.

[23] Leonard, A.: *Energy cascade in Large-eddy simulation of turbulent fluid flows.* Advances in Geophisiscs, A(18):237–248, 1974.

[24] Lilly, D. K.: *A Proposed Modification of the Germano Subgrid Scale Closure Method.* Phys. Fluids A, 4(3):633–635, 1992.

[25] Marble, F. E. and J. E. Broadwell: *The coherent flame madel for turbulent chemical reactions.* Project SQUID Technical Report, Purdue University, 1977.

[26] Moin, P., K. Squires, W. Cabot and S. Lee: *A dynamic subgridscale model for compressible turbulence and scalar transport.* AIP - Physics of Fluids, 3(2746), 1991.

[27] Peterson, B., E. Baum, B. Boehm and A. Dreizler: *High-speed PIV and LIF imaging of temperature stratification in an internal combustion engine.* Proceedings of the combustion institute, page 3653, 2013.

[28] Pignone, G. A., Vercelli U. R.: *Motori ad alta potenza specifica (italian language).* Giorgio Nada Editore, Vimodrone (Milano), 1995.

[29] Poinsot, T. and D. Veynante: *Theoretical and Numerical Combustion.* Edwards, Philadelphia, 2005.

[30] Pope, S. B.: *Turbulent Flows.* Cambridge, UK: Cambridge University Press, 2000.

[31] Pope, S. B.: *Ten questions concerning the large-eddy simulation of turbulent flows.* New Journal of Physics, 6(35), 2004.

[32] Pracht, W. E. Journal of computational physics, 17(132), 1975.

[33] Reynolds, O.: *III. An experimental investigation of the circumstances which determine wether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels.* Phil. Trans. R. Soc. London, 174(935), 1883.

[34] Reynolds, O.: *IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion.* Phil. Trans. R. Soc. London A, 186(123), 1894.

[35] Sick, V., D. Reuss, C. Rutland, D. Haworth, J. Oefelein, J. Janicka, T. W. Kuo and X. Y. Freitag: *A common engine platform for engine LES development and validation.* LES4ICE, 2010.

[36] Sjunnesson, A., P. Henriksson and C. P. Lofstrom. AIAA, 1992.

[37] Smagorinsky, J.: *General circulation experiments with the primitive equations - I. The basic experiment.* Monthly Weather Rev., 91(3):99–164, 1963.

[38] Spurk, J. H.: *Stroemungslehre, Einfuerung in die Theorie der Stroemungen.* Springer-Verlag, 1987.

[39] Thompson, P. A.: *Compressible-fluid Dynamics.* McGraw-Hill, New York, 1972.

[40] Williams, F. A.: *Combustion theory.* Addison-Wesley, Menlo Park, 1985.

[41] Zel'dovich, Y. B.: *On the theory of combustion of initially unmixed gases.* Zhur. Tekh. Fiz., 19(10):1199–1210, 1949. In Russian: K Teorii Gorenia Neperemeshannykh Gazov; Engl. translation: NACA Tech. Memo. No 1296, 1951.

Lebenslauf
MATTEO GHELFI

Name: Ghelfi Matteo
Geburtsdatum und -ort: 08.06.1984, Ferrara, Italien
Anschrift: Reuchlinstr. 29/1,70794, Filderstadt, Deutschland
Telefon: +491603485601
Email: matteo.ghelfi@gmail.com
Familienstand: ledig

Studium

04.2013-heute	Berechnungsingenieur beim TWT-GmbH Science & Innovation (Neuehausen auf den Fildern). Aerodinamik und Kühlsysteme berechnungen.
10.2009-03.2013	Doktorarbeit an der Technische Universität Darmstadt. Fachgebiet: Energie- und Kraftwerkstechnik. Thema: Large eddy simulation (LES) in internal combustion engines,
2008	Master in Vehicle Engineering at the University of Modena and Reggio Emilia. Note: 108/110
2006	Bachelor in Mechanical Engineering at the University of Ferrara. Note: 99/110

Schulische Ausbildung

2003	Ausbildung zum Techniker: Institute of Industrial Nicholas Copernicus Ferrara. Note: 80/100

Berufliche Ausbildung / Praktika

Dez 2009-Okt 2011	Soft skill Kurse an der TU-Darmstadt. Module: 1- communication and team work, 2- presentation training, 3- supervision of BSc and MSc thesis, 4- career development.
Jun-Dez 2008	Praktikum für die Masterarbeit in Ferrari Spa Entwiklung Abteilung. Thema: 1D-3D Analyse von Motorkühlsystem und by-pass Ventile. Masterarbeit: "Analysis of the combined 1D-3D fluid dynamics of a cooling system and a thermostatic valve for high-performance engine", in italienischer Sprache.
Mär-Jul 2006	Praktikum für Bachelorarbeit in Assali Emiliani Carraro Group, Softwareentwicklung für Lagerverwaltung und Controlling. Bacheleorarbeit: "Application of 6-Sigma approach to management of stocks", in italienischer Sprache.